Issue5 July 2006 | Blenderart

blender art MAGAZINE

Blender learning made easy

Modeling Techniques & Blender Scripts

- Blender Material Library
- Gen3 - Tree Generator
- UV Mapping Techniques
- Blueprint Setup
- Spin Modeling

EDITOR/DESIGNER
Gaurav Nawani gaurav@blenderart.org

MANAGING EDITOR
Sandra Gilbert sandra@blenderart.org

WEBSITE
Nam Pham nam@blenderart.org

PROOFREADING
Kernon Dillon

COVER IMAGE
Jason Pierce

WRITERS
Mariano Hidalgo
Edouard de Mahieu
Sergey Prokhorchuk
Roja
Andreia Leal Schemid & Zag
Diego Restrepo París

Click image/page no to jump to article.

Sandra Gilbert
Managing Editor

As I browse the forums, the one question I often see popping up from newcomers, is always some variation of “What is the best way/method/technique to model . . . (insert relevant object here)? Or even more generic question like “What is the best way to model?” I remember my own frustration in finding the perfect answer to that question and sympathize with newcomers who are currently trying to find the perfect method and workflow.

The short answer is . . .
“whatever method works best for your purpose and personal workflow” Not real helpful, now is it? Unfortunately, the long answer is not much better. The reason being that is time, experience and massive experimentation is generally required in order to decide on an object by object basis which method works best for you. Not exactly something a newcomer wants to hear.

Well in this issue we will shed a little light on which techniques to use and when, keeping in mind that anything can be created using any of the techniques available in Blender, but some techniques lend themselves more readily to certain types of models. And while this is obviously helpful to the newcomer, even the more experienced among us will benefit from being reminded that there is always more than one way to model an object.
As much as we rely on favorite modeling techniques, we often find some tasks are either too daunting, repetitive or just a pain to complete manually. Enter Python scripts to the rescue. Python scripts allow greater flexibility and add useful functions and features to the Blender workflow. This issue we will explore some of these wonderful little scripts and show just how they can make your life easier.

sandy@blenderart.org

Izzy's Talk on Modeling

Before you can decide which method of modeling works for you, it helps to know exactly what options are available and how they work. Blender offers a great set of core tools to make whatever method you use as productive as possible. Here is a list of methods and tools for you to consider as you start your next project:

Box Modeling

Generally the easiest method when you first start learning. You'd be amazed by how many things get their start as a cube. Start by adding a pre-made primitive (usually a cube, but can be any of the other primitives i.e. Sphere, Cone, Tube etc). Then extrude, delete edges and vertices as needed to achieve your desired shape.

The use of Box Modeling is preferable if you are able to conceive the desired shape in lesser amounts of detail in the beginning. Basically, you start by creating a basic box, then defining details as you carry on.

Vertex Pushing

Also referred to as point-to-point modeling. This method requires more patience but allows for greater precision, especially when creating smooth Edge Loops for animation. Start with one vertex, extruding and/or adding vertices one-by-one, adjusting and refining your shape until your model is completed.

This method is in contrast to the box modeling technique. It suits people who are able to conceive the shape and details in advance, instead of adding details later.

Curves

Allows you to create smooth shapes and complex geometry. Great for logos.

Nurbs

One of the more difficult methods of modeling. You create patches that are then stitched together to create smooth seamless models. Great for organic modeling.

Metaballs

Much like working with clay. You add additional metaballs with varying influence (can be positive or negative influence) to build up your model.

Tools that can be used in conjunction with most modeling methods.

Extrude: **[E key]** Vertices, Edges and Faces can all be extruded to build your models.

Knife: **[K key]** can be used to create additional Edges and Faces, useful in adding more detail once your basic model is created. Vertex/Edge/Face Select: (Toggle buttons on the 3D window Header Bar [looks like dots/line/triangle]) Useful for selecting just the area you need.

Loop Select: **[Alt + MMB]** allows you to select an entire loop for modification.
Subsurf: (Subdivision Surfaces found in the Modifiers panel of the Edit buttons window) Subsurf smooths the hard angles of your model, maintaining low polygon counts while modeling.

There are actually many more useful tools included within Blender, but these are the ones used most often. More information on Blender's tools and methods, as well as various tutorials can, of course, be found in the Blender wikibook found at:

http://mediawiki.blender.org/index.php/Main_Page

Artweaver 0.4 released

A newer version of Artweaver 0.4 was released on July 18th. The new update brings a lot of changes to Artweaver. It now sports a much improved interface for handling brushes. Also, there are a few new tools in this release: Eraser, Stamp and Perspective Grid.

The arrangement of the Tools and Options dialogs present a clean interface compared to earlier versions.

Also making an appearance are a few new filters such as Lighting Effects, FishEye Lens, and Drop Shadow. Now, saving and loading of selections is also possible. In addition to these features, a lot more are available (including a considerable amount of bug fixes). Read the Release Log for details.

You can get the latest copy at http://artweaver.de

Fig. Artweaver 0.4 Improved interface

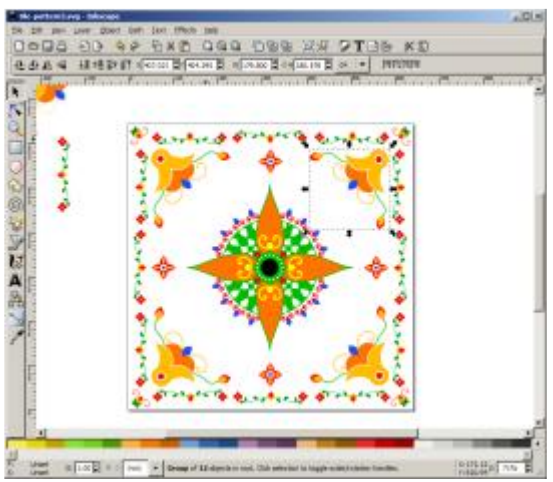


Fig. Inkscape 0.44

Inkscape 0.44 release

Earlier this month, the Inkscape team released one of their most long-awaited releases. It was important because they have improved lots of issues that had hounded Inkscape in the past. The most prominent being the sluggishness when editing objects at higher zoom levels. This release brings a refreshing improvement in speed when handling larger and more complex objects. The node editing is about 30% faster and object movement on the canvas has also gotten a boost in speed. Using the calligraphic pen is also faster in certain cases.

This update also brings numerous improvements to the Inkscape UI. Noticeable improvements are the Color Palette Bar at the bottom and an inclusion of more options on the top toolbar. Also implemented is the Layers dialog box. The overhaul of the Preferences dialog box makes modifying preferences a breeze. And, there's the addition of Keyboard configuration.

There are plenty more improvements and lots and lots of new features packed into this release. Go to http://inkscape.org and download the latest release to learn more about all the new features.

BLENDER LIBRARY SCRIPT

- by Mariano Hidalgo

Level: Intermediate

'Blender Library' is a python script for Blender intended to provide an easy way to store, manage and retrieve frequently used items; such as materials, textures, objects, etc. It also tries to define a standardized way of sharing stuff between the rest of the Blender community, with built-in import and export functionality (This feature needs a full Python installation for the compressed support, please read below for workaround).

Since the script is merely an interface built on top of the current Blender appending system **[Shift + F1]** it will be most likely compatible with all future enhancements in upcoming Blender releases. As an example the material library was coded before the developing of the new Material Nodes and without changes in the script it can now store materials with nodes.

Setting it up: Grab the script from here. *http://uselessdreamer.byethost32.com/scripts/blender_library.py*

Once downloaded, just place the script in your Blender's scripts folder and start or restart Blender (you can also use *(Scripts >> Update Menus)* to refresh your scripts). The script will now be listed in the (*Scripts >> Object>> Blender Library)* scripts menu.

On the first run it will ask you to choose a location for the library folder. Please try to choose a directory not too deep in your file system, this will avoid possible problems with Python's path handling, even though users report that this is no longer an issue in latest versions of the script.

Fig1. Initial dialog.

That's it! your library is ready to be filled with items.

Until you get comfortable with the script there is a context help feature, and if you leave it open it will constantly show tips and advice as you open menus and navigate the forms. Once you have mastered everything, you can close it by Just clicking on the little button on top with an interrogation mark. So the script window gets smaller and don't eat that much space on your screen.

First steps

The script starts in browsing mode. At the top, there are three drop-down menus. The uppermost one is the currently browsed library, the other two are categories and sub-categories. Sub categories are optional, but they become useful once you start filling the library with

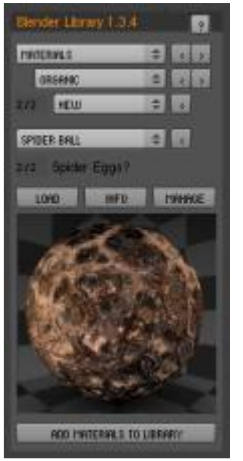


Fig2. The Material library in browsing mode.

lots of stuff. The little arrows on the side of the menus allow to quickly browse the content.

Now you can start experimenting with the script. Either add some stuff you have (read below), or go to Blender Stuff (http:\\blender-stuff.byethost32.com) and download some ready to use items. Downloaded items can be imported one by one or in a batch mode. If you downloaded several items, just choose "Import a Folder" accessible from the 'Right Click' Menu and point it to the folder where the items reside.

Once you reach a category with items on it, a row with extra buttons will appear, along with a preview image and a drop-down menu to quickly jump to other items in the category.

The buttons above the preview are:
• **The LOAD button**. It will pop a menu with different loading options, depending of the type of item.
• **The INFO button**, to display a menu with extended information about the current item. If you **[Ctrl + LMB]** in item's name or description you can change them. Author's information is displayed below and can't be changed for the moment.

To manage the content, the script provides several options in the MANAGE menu. You can rename, delete or move the content to other categories. You can also import a custom preview image to replace the current one.

After these fist steps, it will be cool to open up the preferences dialog, when you can enter your personal information. Then, when adding items you can use the FILL button in the ADD form to automatically enter your data, saving you some typing every time you make new item.

Fig3. Adding stuff to the library

Adding Items

To add your own stuff to the library, click in the Add button at the bottom of the script. A form will show up. Selecting the desired items works a bit different depending on which library your are on.

The 'Object' and 'Lamp' libraries can store several Blender's datablocks in the same item, so you are not limited to store single meshes. You can store and share a full character, complete with props and armature, an even shape keys attached. By storing several lamps in the same item you can create lamps sets to achieve reusable lighting environments or on click-to-setup fake global illumination. Choose the desired objects or lamps on the 3D View and then type a name for the item in the SET NAME: label.

For the rest of the libraries you select the desired item from a menu accessed from the right pointing arrow next to the NAME label. In this case the label will be auto filled with chosen datablock's name. Despite this, all related stuff to the chosen item is also added. If you add a material, it's textures and Ipos are also added, and so on.

For some libraries you can turn on the ENABLE RAY button, in case you want a ray traced preview image, and for materials you can also choose the shape of the preview object from a drop-down menu.

Now its time to click the Add Item button to finish the process. The script will warm you about the need to reopen the current blend file after adding. This is a small limitation (which may disappear in future versions) so please save your blend before adding stuff to the library. If you accidentally forgot to save and lose work, you can search for the quit.blend file, that is a Blender feature to recover blend file, which Blender automatically creates for safety.

After reload, run the script again and the item you have just added will be there for quick reuse.

If you plan to share your items, and to protect your rights as a creator it is possible to bundle a License Text with every item you add to the library. To do this, have your license text opened in the current .blend and then select it from the drop down menu. When people load your item, the script will warn them about the presence of the license and its text will load into the project.

Exporting Items

If you want to share your creations with other people, use the Export Item option in the Manage menu. You can export compressed TAR packages only if you have a full Python installation.

As an alternative you can run a search on google or you favorite search engine and look for these two files: shutil.py and tarfile.py; download and place in your Blender's folder. Now you can export and import compressed items.

When you export compressed items the script gives you a JPG image along with the TAR package. The image is not needed to import the item back, rather it provided snapshot which you can show as preview in the forums or websites when you may be posting your stuff.

If you are exporting uncompressed items, then the three resulting files are needed and you should distribute them all. It would be a good idea to pack them manually as a TAR file by hand.

Fig4. Exporting items

The Libraries

Lets take a look at some of the libraries.

The Material and Texture Libs The material library is the star of the show. It can store regular materials, materials with nodes and even materials with IPOs attached. If the material uses Image' textures, the image files will be automatically packed. The texture library is cool for storage of procedural settings and environment maps. As a plus, textures of type Blend can be used to store color bands.

The (Game) Logic Lib This is intended for game developers and is a quick way to store logic bricks. This way you can store your keyboard mappings or the bricks needed to make a pan and rotate view tool and quickly reuse them later. To store game models is better to use the Object library instead, the TEXFACE button will allow you to get a right preview for UV textured models.

The Scene Lib All the time people download from the net blends with examples and concepts, usually with a single scene. With this lib you can easily catalog that files and also its a good place to store several assorted items you may want to load at once.

The Particle Lib.
Can store static (hair, fur, grass) or animated particle systems, the later with attached force fields and deflectors.

For the later to work, you need to name each force field and deflector beginning with the name of the particle object, plus a dot, plus their name

(for example: PARTICLEOBJECT.FORCEFIELD, PARTICLEOBJECT.DEFLECTOR.001)

The Armature, Action, Pose and IPO Libs. Favorites for animators, they store animation related datablocks and even single poses.
The pose library currently doesn't store those poses which use attached Target IKs, but this may change in future versions. The automatic IK works just fine.

Fig5. A complete animated particle system just one click away

The World Lib

It is possible to store animated worlds, a very cool feature for people using Blender for advertising or multimedia design. In the latest version of the script, you can also choose to just load Particle or Mist settings, to merge with your current world.

Extra Functionality

Once you start collecting items, you'll need the Search function, which can be accessed from the 'Right Click' Menu. It will let you search a specific library for matching items both by item's name and author's name. Once found, the matching items will be shown just like in the browsing mode, ready to be loaded in your current project. Also remember that most used items can be added to favorites for quick access and they will be listed at the bottom of the LOAD menu. To remove a favorite just **[Ctrl + LMB]** on it's name.

Round Down

Current version of the script is 1.3.4. For future versions the small implementational quirks may be gone and hopefully there will be no longer need to reopen the .blend after adding stuff. Also, maybe giving the script a custom space will be a cool idea.

Fig6. Looking for materials in the search panel

You can visit the Blender Library thread at BlenderArtists forum to post doubts queries or feature requests. You can also take a look at my site http://uselessdreamer.byethost32.com for more scripts.

Hope the Blender Heads around the world find this tool useful and we'll soon start creating a huge community of people sharing all kind of great Blender content.

– See ya around
Mariano Hidalgo

Mariano Hidalgo A.K.A.
uselessdreamer

I'm a graphic artist, designer and musician from Buenos Aires, Argentina.

Sometimes I code for fun and I manage to get away with it ;)

GEN3 TREE MODEL GENERATOR SCRIPT

- by Sergey Prokhorchuk

Level - Intermediate

Manually creating realistic tree models is a hard task in 3D/CG. To simplify this task, various plug-ins and programs, which can be used to create 3D plant models automatically, were created. These programs allow the users to describe plant structure by inputing some parameter settings which are used to generate the 3D model.

For Blender users, [LSystem] and [Arbaro], are the two common plant generators. [LSystem] is a python-based script for Blender which generates plant models based on the "L-system" algorithm - that's why the name "Lsystem". [Arbaro] is a stand-alone Java-application, which generates plants using the model described in [Weber95].

Gen3 is a relatively new tree generation plug-in/script for Blender. Its first version (v0.1) was released on June 17th of 2006. To generate tree structures, it uses models described in [Weber95](the same model that Arbaro uses).

The amount of included parameters describing tree features are vast. Broadly, these parameters allow the user to describe the trunk and branches sizes, their split probabilities, branch locations relative to each other and so on. Currently, the interface is not user-friendly and may scare off new users, but the UI re-implementation is a possibility for future releases.

Installation

To install Gen3, you should download the latest version of the script from here[Gen3]. (At the time of this writing, the latest version available is 0.5) Unpack the downloaded archive

Fig1. Script access menu

to a temporary directory and copy the file gen3.py and the “gt” sub-directory to your Blender script directory (Which can be located in the Blender installation path). It is highly recommended to ensure that the versions of the 'standalone Python Installation' and the Blender Python versions are the same and were compiled with the same build options. This requirement is necessary as Gen3 uses some modules which are part of the Python library.

After copying the files, you are ready to use the script.

Usage

You can access the Gen3 script via *(Scripts>>Misc>>Gen3)* see fig1. If you do not see "Gen3" entry in the Scripts menu then select *(Scripts>>Update Menus)* - it will update the list of available scripts.

After launching the script, you will see all the controls which allow you to control parameters affecting the model of the tree. You can find complete description of all these parameters in the [Weber95] paper.

There are several buttons at the bottom part of script's GUI, of which the main button is the "Generate" button. Upon clicking this button, the script starts generating a 3D model. To make things easier for you, there are some buttons with the name of tree types on them. These buttons will fill up predefined parameters for the selected tree which can be a good place for you to start experimenting.

For a quick start, you can press the "Quake Aspen" button and then press the “Generate" button. After a while (time will vary depending on your computer configuration), you will see a generated tree model in the 3D view window.

Importing parameters from [Arbaro]

Since Gen3 uses the same algorithm as Arbaro, it is possible to import tree parameters generated by Arbaro. Notice that the import feature is available only if the Blender python interpreter has access to the xml.dom.minidom module, which is a part of Python library.

If it is installed properly then, you will see the "Import" button right under the "Generate" button. To import parameters, simply press the "Import" button and select the Arbaro generated XML file to import parameters. One thing that must be noted is that some Arbaro-specific parameters are not supported in Gen3, so the generation of trees may differ.

Custom leaves usage

Gen3 allows the use of any mesh/object as a template for leaf generation. To use a custom mesh/object as a leaf template, you must add a prefix "Leaf" to its name ("LeafSmall" for example). To refresh available leaf templates for the script, press the "P" button located near the "LeafShape" drop-down list.

After refreshing, all meshes which have the "Leaf" prefix in their name will be available in the "LeafShape" drop-down list. Select the needed object in the list and it will be used during the generation of leaves.

Credits

I'd like to thank Jason Weber and Joseph Penn for their excellent paper on plant structures. Also, I want to say thanks to all members of the http://blenderartist.org forum for their responses, advices and suggestions. Thanks to all who've helped in the development of Gen3.

by- Sergey Prokhorchuk

Fig2. Gen3 UI explained

Links

[Arbaro] http://arbaro.sourceforge.net
[LSysInfo] http://en.wikipedia.org/wiki/Lindenmayer_system
[LSystem] http://jmsoler.free.fr/util/blenderfile/images/lsystem/lsystem.htm
[Gen3] http://geocities.com/bgen3
[Weber95] Jason Weber, Joseph Penn, "Creation and Rendering of Realistic Trees", Proceedings of ACM SIGGRAPH 1995: pp. 119-128.

L-System

L-system is a method of describing self-similar structures, it can be vaguely compared to fractals. L-system definition consists of:

1. Set of symbols from some alphabet.
2. Production rules.
3. Meanings attached to each symbol from the alphabet.

The main part of L-system is a string rewriting module, which applies production rules to some initial string. The Rewriting procedure is applied to the string in a recursive manner. After some iterations, resulting strings can describe very complex self-similar structures. Since many plants have self-similar properties, L-system can be used to describe them. Note, that L-system can also be used for generating many other complex structures and therefore is not restricted to just plants.

Jason Weber and Joseph Penn's tree generation model

Jason Weber and Joseph Penn's tree generation model information box ---
This model, unlike the L-system, is oriented specifically for 3D plant generation. In this model, trees are described under trunk and several levels of branches. The model describes parameters and formulas which are used to generate only tree models. Some parameters apply to models in general, some to trunk, branches or leaves only. The parameter settings are rich enough for complex tree families.

These settings allow configuration of sizes and orientations, trunks and branches, shapes, probabilities of stems splitting, leaf shapes and so on. Parameters describing trunk flaring, lobes and tree shape pruning are also present. Since this model was created specially for tree model generation, it can be understood and modified more easily than models based on the L-system. A full model description can be found in [Weber95].

Sergey Prokhorchuk

I am a software developer at ViaSoft Ltd, Ukraine. Since I have some experience in C/C++ and Python programming, I try to use it in different areas of 3D-graphic. Other computer-related areas I'm interested in are: programming theory, artificial intelligence and cryptography.

MODELING RUPERT THE 'EVIL MONKEY'

- by Sandra Gilbert

Level - Intermediate

Character modeling is one of my favorite activities in Blender. I get the biggest kick out of creating a character and watching it come to life. Since my style of modeling leans toward toon-style characters (obviously I spent way too much time watching Looney Tunes as a child), and I admittedly don't draw as well as I want, I find most of my inspiration for characters in the toy aisle of my local department store.

Children's toys make wonderful reference models for characters. They provide a wide range of choices from colorful fantasy types to realistic characters and everything in between. Even if the character you had in mind isn't to be found in the toy aisle, most likely you can find something close to get you started and then you can alter it from there to match what you had in mind.

In this tutorial, I'm going to show you how I went about modeling an evil monkey. The reference model I'm using comes from a very popular line of toys by Hasbro, "My Littlest Pet Shop". They are very simple characters based on a wide range of animals. (And yes, I have most all of them, for modeling reference of course)

This tutorial is a beginner to intermediate range tutorial. Although I do assume that you can at least get around fairly comfortably in Blender, I will do my best not to skip steps that I think are common knowledge. That being said, let's get started. Make sure you have a clean workspace, if you normally have a cube or plane by default, delete it.

First we need to get our reference image into Blender

For my purposes, I took a front and side photo of my monkey, who I have named "Rupert".

In Photoshop (GIMP can be used just as well, and or any other image program that you feel comfortable with), I put both images side-by-side, making sure that the size of each matched up.

Next, let's get him into Blender. In the front view [Numpad 1], go to (View>>View Background). A dialog box will pop up (fig 1). Browse to where you have stored your image of Rupert.

I like to work in a split-screen layout, with one screen for front view and one for side view, so I repeated the previous step for my side view screen [Numpad 3]. You should now have an image of Rupert in both screens to work from.

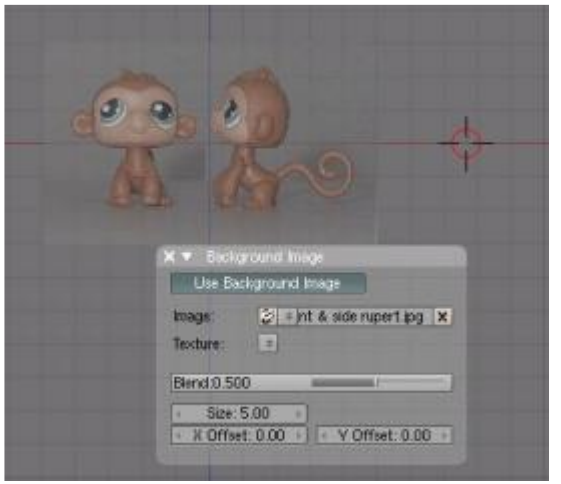


Fig1. Rupert in Blender background

Let's get modeling: Head

There a couple of techniques that could be used here. Although I normally use box modeling, for this model, we are going to explore point-to-point (or vertice) modeling, in an attempt to create nice edge loops for possible animation (LOL although I am fully aware that this probably could be demonstrated better by someone more skilled at it). Also, I prefer to model one half of a character first, mirror it and then add variations to both sides.

Fig2. Adding vertices for head

Head

Go to menu *(Add>> Plane)*, select and delete 3 of the 4 vertices.
In side view **[Numpad 3]** RMB click to select the remaining vertice, **[G key]** to grab and move the vertice to the tip of Rupert's nose.

[Ctrl + LMB] click around Rupert's head to create an outline to work from. See figure 2 for reference.

In front view [Numpad 1], select all the vertices and move them to the center of Rupert's face (fig 3).

Fig3. Positioning vertices

Make sure all vertices are deselected **[A]**, select the vertice right at the spot where Rupert's face changed from tan to brown, then Control LMB click around the tan face area. I started in side view **[Numpad 3]** and then in front view, I moved each vertice until it matched the tan area in front view also. (fig 4)

Fig4. Positioning vertices

Next, **[Ctrl + LMB]** click additional vertices around Rupert's eye, making sure you adjust from both front and side views, also create a line of vertices going from his nose, down across his cheek and around his eye. (fig 5)

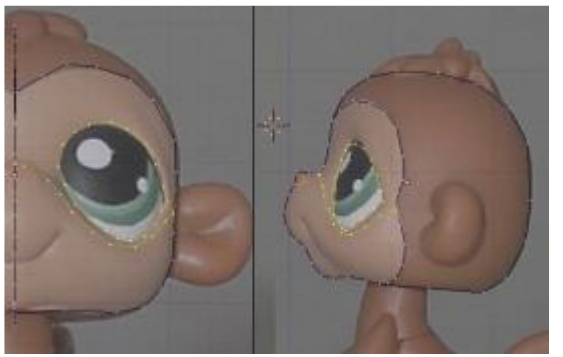


Fig5. Adding eye

Let's finish creating lines for his cheek and mouth area, then we can start filling in the faces in the front of his head. (fig. 6)

Fig5. Adding more details

Filling in of Faces

This part generally takes a little adjusting for me. The goal is to line up all your nice edges into clean 4 sided polygon faces. The best way to go about it is to just jump in and start filling them up. I prefer having more control over this process so I do them one at a time, fixing problem areas as I go.

To fill in faces, select 2 vertices and create an edge between them **[F]**, continue to create edges until you have a closed box (or polygon). Once you have a closed area, select all four vertices and hit **[F]** again to fill the polygon. (fig 7)

Fig7. Filling the faces

Continue creating edges and filling in faces until your model resembles (fig8).

Most likely you will run into same problem as I did. On some of the edge lines, I didn't create enough vertices for every thing to line up nicely.

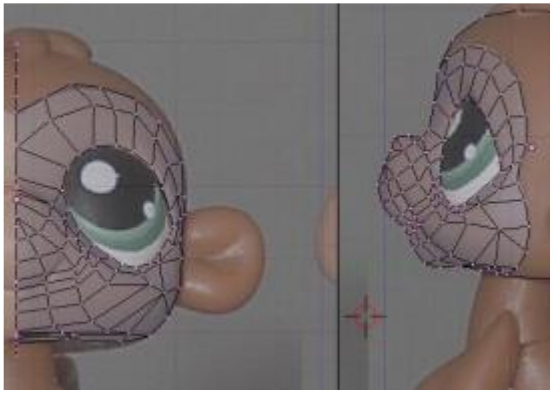


Fig8. Rupert after face filling

I fixed it by selecting 2 vertices to either side of the problem area and hitting *(W>> Subdivide)* in 'Edit Mode' to create additional vertices. For the most part, I managed to create clean four-sided polygon faces. There are a few triangles that couldn't be helped and hopefully won't cause a problem. If they do, we will deal with them later.

At this point, you can check your progress in solid mode **[Z]**. Make sure you have selected your model and have pushed the 'Set Smooth' button in the 'Edit buttons' window. Also, at this point you may have normals facing the wrong way. To fix that, select all vertices while in 'Edit Mode' and press **[Ctrl + N]** to set them all in the same direction. Rupert is looking pretty good at this stage. If you want, you can take a look at him with Subsurf on (remember, Subsurf has been moved to the Modifier stack in Blender2.4x).

Time to create the back of his head

It will be just like creating the edges for his face. Control LMB click to create new vertices following along the reference photo, making sure to adjust from both front and side views to create a nice smooth, rounded head. (fig. 9)

Make sure to model around where the ear will go. If you cover it up now, it will just be added steps to attach the ear later.

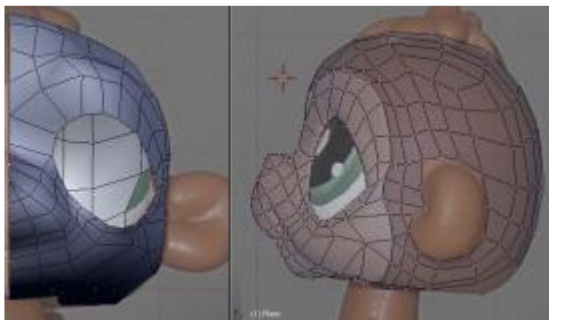


Fig9. Smooth surface

The eyes say it all

Since my reference photo is entirely too cute for an evil monkey, we are going to modify his cuteness factor. And since the eyes say it all, that is what we are going to modify.

Select the vertices just above the eye socket (fig. 10). Grab them [G] and move them forward to create a more prominent brow line and then rotate in both side and front view to give it a more sinister look. Do the same thing with the row of vertices just above the one you just altered (fig. 11).

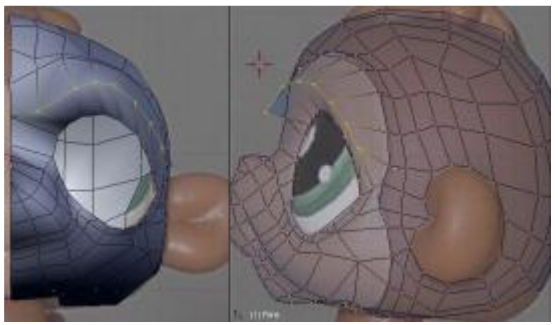


Fig10. Refining eyesocket

Fig11. Refining eyesocket

Next, it is time to refine the eye socket. Select all the vertices around the eye socket. Press **[E key]** to extrude, do not move the mouse and immediately press **[S key]** to scale the vertices slightly smaller than the socket. Press **[E key]** to extrude again, this time move the new row of vertices back into the head and then press **[S key]** to scale it slightly smaller.

At this point you can add in your eyeball. There are many good tutorials on creating eyes, so I'm not going to cover that here. My favorite by far is the Pixar eye tutorial(you can do a google search for it). So use the eyeball of your choice and place it into the socket. (Quick render of eyeball added, fig. 12)

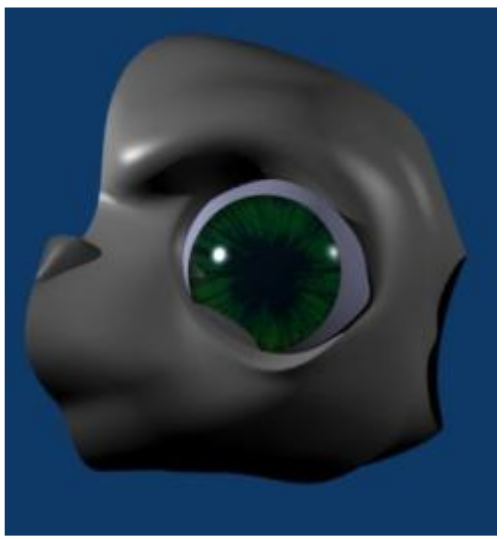


Fig12. Rupert's 'eye'

Ears

You could either model the ear separately and attach it later or model it directly on the head. Seeing as how the ear looks like a modified sphere, I am going to model it separately and then attach it.

First *(Add>> Mesh>> UVSphere)*-{8 segments: 8 rings}. Scale it down to fit the ear. (fig. 13)

Select end pole vertice and the first ring, with proportional edit **[O key]** on {use Smooth Falloff}. Press **[G key]** to move the vertices back into the sphere, then select a couple of the outside vertices and move then back (fig. 14)

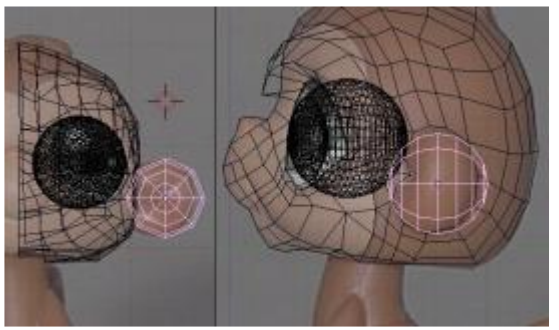


Fig13. Adding ear

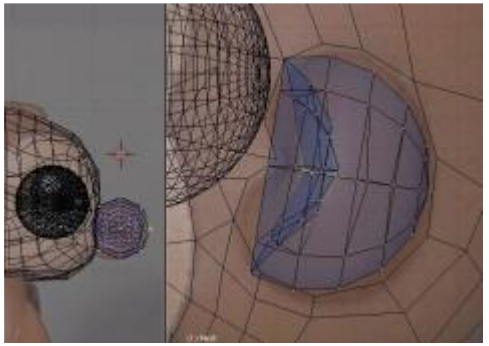


Fig14. Editing ears

Move the ear further into the ear hole, join the ear to the head, keep pressed **[Shift key]** the RMB click to select both objects and then press **[J key]**. Start merging **[Alt + M]** vertices where they line up. Once we have the ear firmly connected, we will delete unneeded vertices from inside the head. (fig. 15) Seeing as how ears are difficult to explain, please look at the included blend file for placement and ear-type tweaking of the sphere.

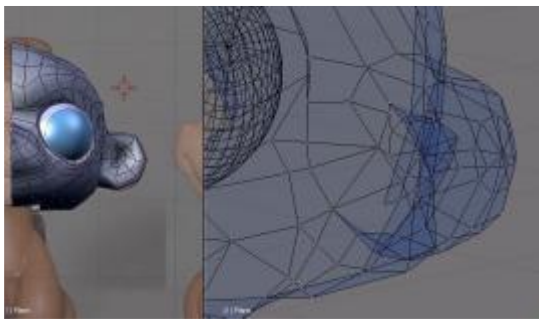


Fig15. Joining ears to the main mesh

Basically you will want to push and pull the vertices inside the ear to create the illusion of folds and ridges. That concludes the building of the head.

Next, we will be working on the body. Rupert's hair will be added as a final cosmetic modification when we have finished.

Let's get modeling: Body

At this point, the extrude tool [E key] is going to be our best friend. We will be extruding out from the bottom of the head to create a seamless flow into the body.

Neck & Body

Select the 4 bottom vertices of the head. This will be where we start creating the neck (fig. 16) and extrude down, adjusting vertices as needed to match up with the image. You will notice that a little adjusting will be needed where it meets the head as well. Continue to extrude the new vertices down, shaping it to the rest of the body, scaling and rotating as needed to create the rounded body shape. (fig. 17)

Fig16. Adding neck

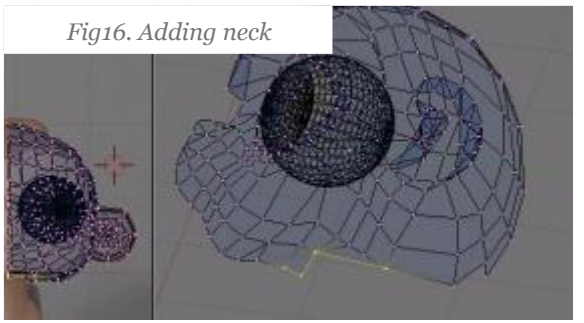


Fig17. Extruded body

At this point you should be ready to model the tail, just keep extruding, following the shape of the tail.(fig. 18) Pause and take a look at your model from all sides, adjusting areas that look

Fig18. adding tail

a little off. I fattened up his body from the front view (select all body vertices and press **[Alt + S]**, drag your mouse horizontally to the screen), you can also grab vertices in the front and back hip areas and pull them out a little to create a more rounded body. Turn on Subsurf to check for smoothness and a preview of how Rupert looks (fig. 19).

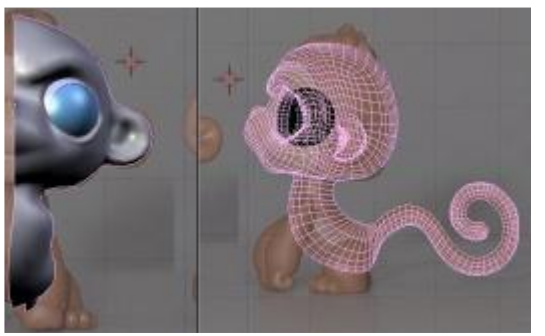


Fig19. Checking mesh in Subsurf

Let's get modeling: Legs
Now we need some legs, we will be box modeling his legs and attaching them later to his body.

Move the selected vertices to line up roughly with where the front leg will go, do the same for the back leg. (fig. 20)

Next, get out of 'Edit mode' **[Tab key]** and then deselect the monkey. We will be building his legs separately and then attaching them later. Go to layer 2 to create the legs (it will be easier to see what we are doing without his body being in our way)

Fig20. Adding legs

Now do (Add>> Cube). Position it just over the front leg of our image. Scale it down so that it matches the image.

You will need to adjust vertices in both the front and side views to get your cube to line up with the image. (fig. 21)

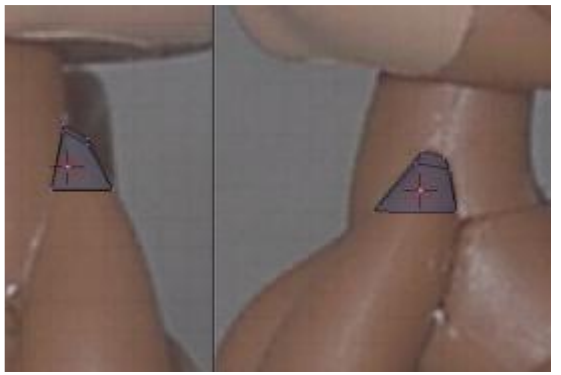


Fig21. Adding hands

Box modeling allows for quick build up of a model. Extrude the bottom four vertices down the length of Rupert's arm, adjusting vertices (scaling and rotating) as needed to achieve the basic shape. (fig.22)

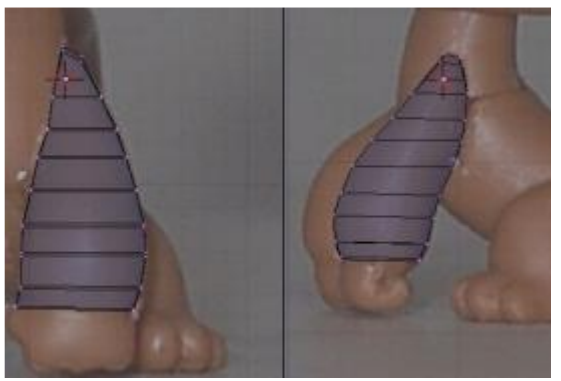


Fig22. Adding hands

I have stopped just short of the hand for the front leg. Since I plan on animating Rupert at a later date and time, eventually I am going to actually model a hand and curl it up using armatures.

You can either model a complete hand or just model the hand curled up as in the image depending on your needs. For the scope of this tutorial, I will just model his hand curled up. (fig. 23)

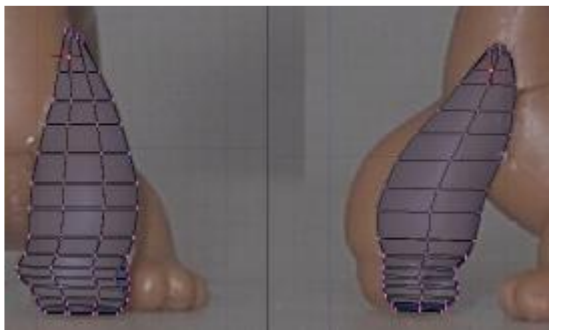


Fig23. Finishing hands

Using the same method, create the back leg. (fig. 24) Remember to create toes when you get to the foot. It is an easy matter to extrude out 3 regions for the individual toes. Although depending on how you extruded from the top of the leg, you may have to make some Loop Cuts **[K key]** to create 3 equal segments at the front of the foot.

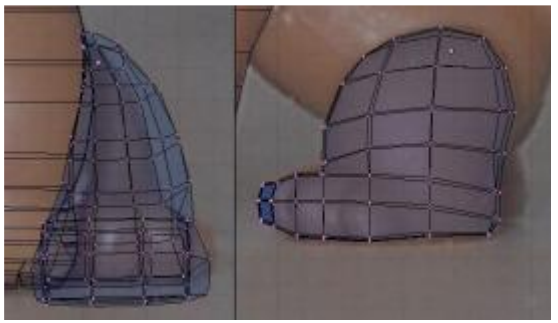


Fig24. Finished legs

Now comes the really fun part (insert heavy sarcasm). We need to attach the legs to the body. First thing we need to do is get the legs on the same layer as the body. So, select both legs and press the **[M key]** (move to new layer) (fig. 25) and push the Layer 1 button. Your legs should now be on the same layer as your body.

With the legs still selected (and while in 'Object Mode'), select the body mesh also and press **[Ctrl + J]** to join all the pieces together.

Switch to 'Edit Mode'. Now comes the fun part. You are going to be matching up individual vertices from the legs to the body. It might be helpful at this point to select everything not needed for this operation and hide it using the **[H key]** (fig. 26)

There is no real easy way to explain how to do this part. Save your file before you start. Start selecting individual vertices from the leg and the closest matching one from the body.

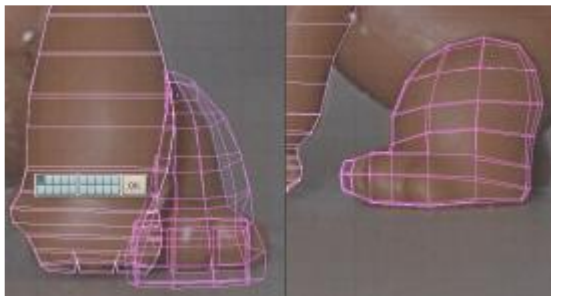


Fig25. Moving to another layer

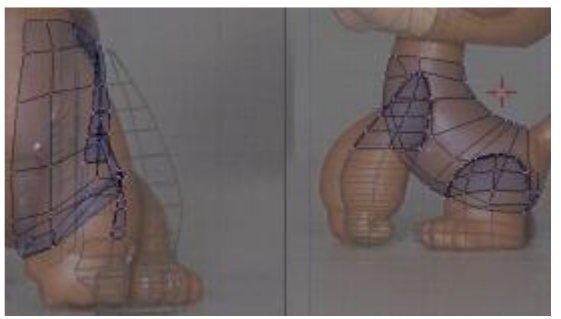


Fig26. Aligin vertices

Press **[Alt + M]** to merge the vertices. You will need to delete some unneeded vertices from both the body and the legs once you are finished. Don't get upset if you have to go back to your saved file and start over a few times. It takes a little experimentation to get it lined up just right.

Once you have attached your legs, turn on Subsurf (it is in the Modifier stack in the 'Edit buttons' window) and look at your model.

Using the Smooth button (found in the 'Mesh tools' panel of the 'Edit buttons' window), select any parts of your model that look a little too sharp or bumpy and then click the Smooth button a few times to smooth those areas. When you have finished smoothing out and tweaking your model, he should look something like this. (fig. 27)

Fig27. The body in 'solid shade view mode'

Putting it together

Well, we have half a monkey, which in itself could be lots of fun, but not overly useful. Time to Mirror and join the two halves to make a whole monkey.

Mirroring

We will be using the 'Mirror Modifier' to create Rupert's other side. First off, while in front view, make sure your object center is on the inside edge of Rupert. This will allow the Modifier to accurately place Rupert's other side (fig. 28)

Fig28. Mirroring

As you can see, the default settings in the 'Mirror Modifier' show a minor problem with putting Rupert together. His edges aren't lining up really nicely at all. So, turn off the display in the Mirror Modifier while we fix him up. (If you still have the Subsurf modifier on, turn it off also)

Select the top vertices of Rupert's body and tail, extrude once from top view [Numpad 7), line it up with the vertices from his head. (fig. 29)

Fig29. Linening up vertices

Next, select all the inside edges of Rupert. Press the **[S key]**, then then **[Alt key]** while moving the mouse horizontally, to scale them into a straight line

Now turn your 'Mirror Modifier' back on, his edges should line up nicer now. You might need to move the object center of Rupert a little closer to his inside edge. (To move an object center, select a vertice on the inside edge of Rupert, press **[Shift + S]**. Select "Cursor to Selection" on the popup menu. Next, go to the 'Edit buttons' window and choose, "Center to Cursor" in the Mesh Panel.)

If Rupert looks all nice and lined up, go ahead and apply the 'Mirror Modifier'. If not, adjust the object center some more until he lines up nicely. Okay, now take a look at Rupert from all angles. There are a few missing faces that we need to fill in (Select surrounding vertices and press **[Shift + F]** to fill all of them). Also at this time, smooth away any weird angles with the "Smooth Button" in the Mesh Tools panel.

Adjust and tweak any areas that just don't look right. Your final model should resemble something close to (Fig. 30). (I did go back and select his tail and scaled it to be a little thinner, but that is a personal choice I will leave up to you.)

At this point you are technically done, now you can put him in your favorite jungle and wher-ever he thinks he needs to be. You can add armatures and swing him on a vine.

And best of all, you get to decide how you will texture him. Will you go for toon shading? Fur? Maybe feathers, just to be different. It's up to you. Personally, I'm just gonna give him a simple coloring, enhanced with Ramp shaders, so that he matches up with some other characters I am already doing.

Fig30. Eeek!! the'evil monkey is alive'

Fig31. Rupert after test render

KITCHEN WORKS 'SPIN MODELING'

By Andreia Leal Schemid

English translation by Zag.

Level - Intermediate

How could we define 3D-Modeling?

3D-Modeling is a set of techniques and methods used to create a complex, real or imaginary model. Using the toolset of a 3D modeling program, we start from simple geometries, combining and organizing them in such a manner that they facilitate the construction of the model.

There is a lot to talk about this issue.Practically, we can imagine one technique for each tool or method that we have in Blender. Therefore, some of them are general-use techniques and can be used in most well-developed 3D programs.

The first question that we have to ask ourselves is, "Which is the best modeling technique to use"?

Well, it depends on your knowledge of that technique and your skills at using it, but any artist or designer will say: "Your model, basically, will define the technique that must be used to achieve the best results". There are no better techniques, in the beginning stages of 3D modeling, than taking an analytical look at your designs and asking some preliminary questions:

Some general questions:
- *Will I need precision in my work?*
- *Will this model be animated?*
- *What level of detail is needed?*

Some specific questions:
- *Will the model be symmetrical?*
- *Should it be assembled from simple parts, pre-made meshes or will it be a one-piece mesh?*
- *Have we some structures or parts that are repeated? Should we use duplication to create these parts?*
- *What style is best for my model?*
- *Will the model be predominantly curve-based or vertex-based?*
- *Can basic forms be used to start the modeling?*

Throughout the modeling process, new questions and new decisions will pop up that must be addressed.

So, how can we survive in this never-ending process without giving up? The answer is ORGANIZATION! Organize your project, your resources, your basic ideas, and most importantly: Know Your tools and your skills.

Spin tool

The Spin tool is simple to use and understand. In this article from Procedural Magazine - the first PDF Magazine in Brazil, KHA (nickname of Lady Andreia Leal Schemid), we will discuss some difficulties related to this method and the solutions to deal with good work in "Spin Modeling".

In this article, we'll construct a model of a wine glass, utilizing the Spin technique. Spin is a very simple tool that allows us to quickly remake all of the mesh or adjust it by moving some vertices when necessary. This could reduce a lot of modeling work.

Furthermore, the resulting mesh will be totally symmetrical. The Spin tool creates a 3D mesh by rotating a shape around an axis.
(blue line on fig 2).

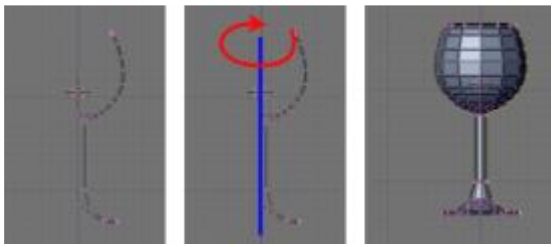


Fig2. Sping modeling

The first thing we need is a shape to work on. In the front view **[Numpad 1]**, add a plane **[Space bar >> Add >> Plane]** as you see in figure 4.

Fig4. Adding a plane

As we are in the Edit mode, we will delete three of the four vertices of the plane.

Use **[Shift+RMB]** to multiple select vertices, and **[X Key]** or **[Del Key]** to remove the selected vertices. See the figure 5.

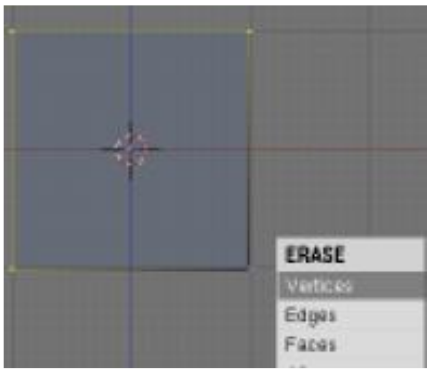


Fig5. Removing the vertices

Now select this vertex with the **[RMB]** and move it to a vertical line **[Ctrl + LBM]** click (the central blue line) Image6-7. in the front view.

Fig6-7. Extruding the vertice

Start the Shape drawing

With the vertex selected, press and hold **[CRTL]**. With the left mouse button, click and generate more vertices in a sequence. You could use Grab **[G Key]** to move the new vertices and create a shape as you see in the figure 8.
Anytime you make mistakes, use **[U Key]** to undo the error and keep on working.
Remember, that when we are modeling an empty object,

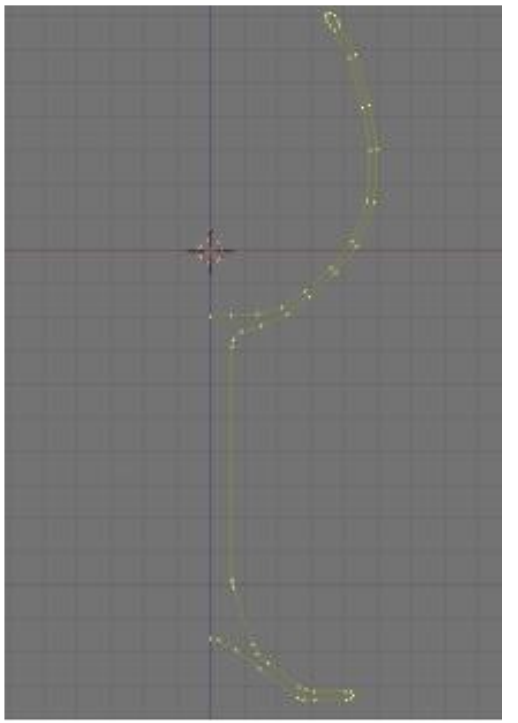


Fig8. The shape of the glass

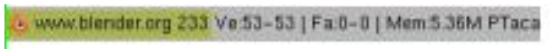


Fig9. Number of vertices in the shape

We have to draw the external and the internal lines to obtain the thickness. Another important point is that the starting vertex and the final one must line up vertically.

The number of vertices may vary; I have used 53 vertices in total. As a rule, use more vertices in the more curvaceous parts of the shape.

The Position of the 3D Cursor

The 3D Cursor position will define the position of the rotation axis. Then, with the shape finished, move the 3D Cursor over the vertical blue line. To do that, click the [LMB] on the blue line, press **[Shift+S]** and in the Snap pop-up, select **[Cursor>> Grid]**. See fgure 10.

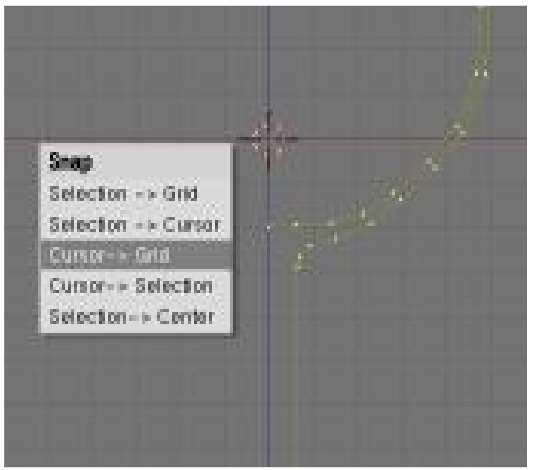


Fig10. The 3D Cursor must stay in the same line of the start and end points of the shape

Also, be sure that the 3D Cursor is on the Object Center in the top view **[Numpad 7]** as you see in the figure 11.

Fig11. The top view of 3D Cursor position

Changing the View to Rotate

The current view will define with axis that will be used for rotations. Then go to the top view **[Numpad 7]**. Press [F9] and in the "Mesh Tools" panel of edit buttons, set the parameters for the Spin tool. (figure 12)

Fig12. Settimg the 'spin tool'

Define [Degr] to 360, to have a complete turn of the shape.
The number of Steps to 12.
The number of Turns to 1.
Click [Spin] to start.

You might prefer to make your mesh with more Steps. To do that, just press [U]. Then, modify the Steps number and [Spin]. I prefer to keep it simple and, afterwards, use Subsurf to smooth things out for the Render. Another important thing is to verify that the [Clockwise] button is pressed. Our mesh will be automatically formed.

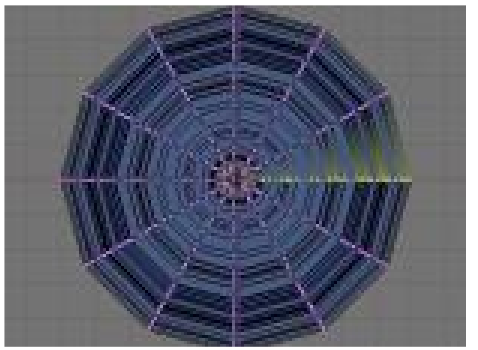


Fig13. The created mesh after 'spin'

Observe that the start shape is still selected, and the final shape of the rotation and initial one are coincident. But, the mesh is not a closed mesh.

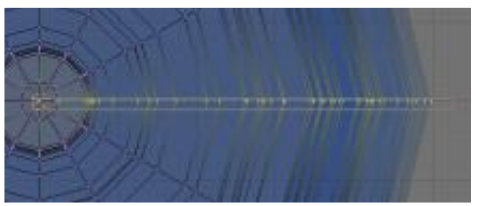


Fig14. Close up of selected shapes

To close it, we need to select the vertex in the initial and in the final shapes using the box selection tool [B Key]. Pay attention to not select the pink vertex too. With all the vertices selected, verify the total number of selected vertices. In my example, I have 128 which is greater than 106(53x2). see the figure 15

Fig15. Wrong number of vertices

What's wrong? Maybe the start and end points could not be perfectly aligned in a vertical line and then we create a circle where we need just a point. See figure 16.

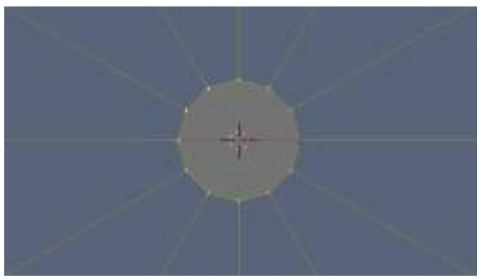


Fig16. The start and end vertices are not aligned and the rotation generates a circle

To solve this, select exclusively the vertices in the start and end shapes. You should have selected 106 vertices in total. De-select the vertices that form the circle (figure 17) and be sure that the other vertices are still selected (figure 18).

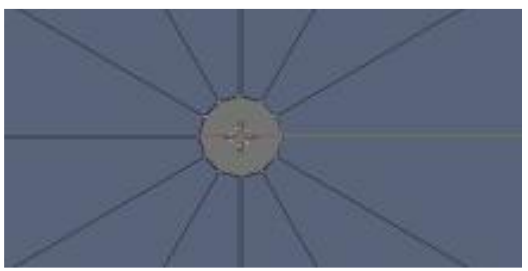


Fig17. Only start and end vertices selection

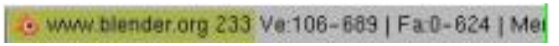


Fig18. Only start and end vertices are selected

We see a pop-up menu informing how many vertices were removed; in this case, 53 - see figure 20

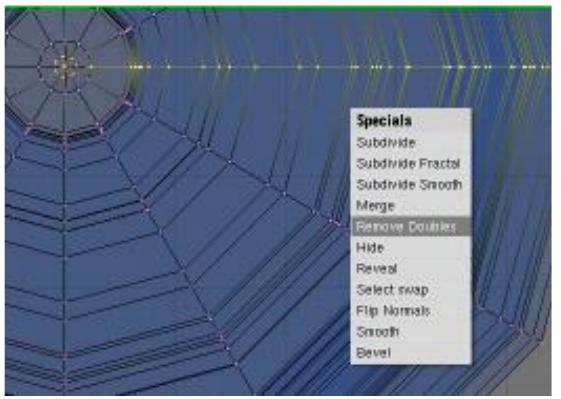


Fig19. Joining the vertices

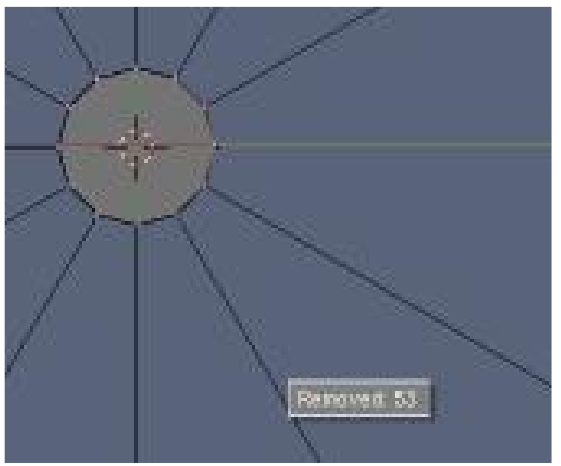


Fig20. Now the mesh is properly closed

Now, select only the vertices forming the circle and look at the number of vertices selected. In my example, there are 24. See fig 21.

Fig21. Count of vertices

In the front view,[Numpad 1] we can see that these vertices belong to two different circles. See figures 22 and 23.

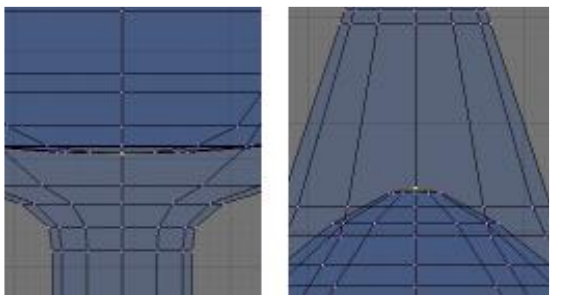


Fig22-23. Two series

We want to keep the vertices in the base of the glass selected. The vertices located inside the glass will be deleted. Use the box selection tool [B Key] and drag-select the vertices in the top part of the glass while keeping the keyboard button [Alt] pressed. This will de-select those vertices.(figure 24)

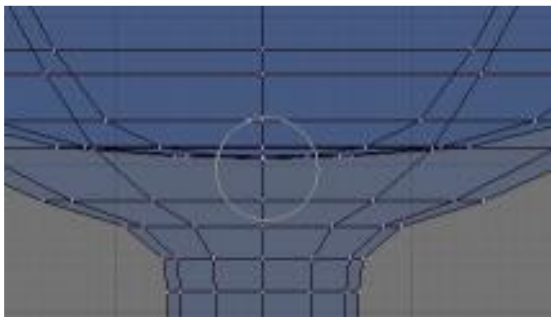


Fig24. Deselecting vertices inside the glass

Go to the top view [Numpad 7] and we see that only the circular vertices in the base are selected. (figure 25)

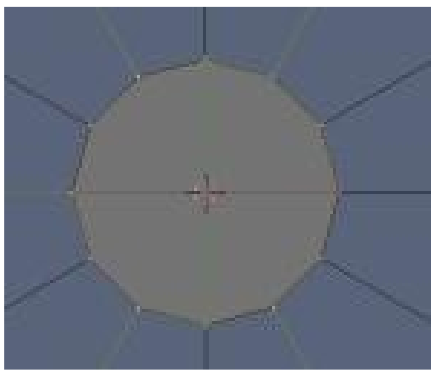


Fig25. The circle of the base selected

Let's join these vertices. Press the [S Key] to scale the circle, moving the mouse until the vertices overlap. (figure 26)

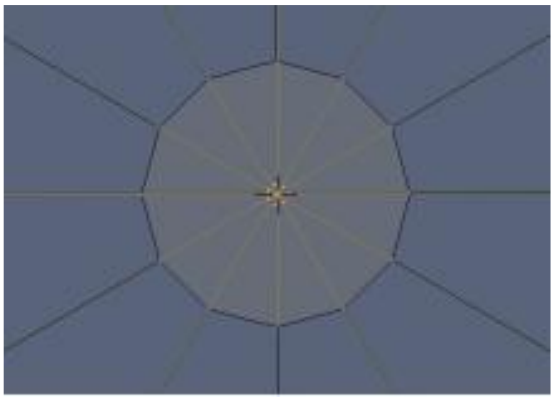


Fig26. Joining the vertices

We now need to remove any duplicate vertices found in the selection. To Remove Doubles, press the [W Key] and then select Remove Doubles in the pop-up menu. (figure 27)

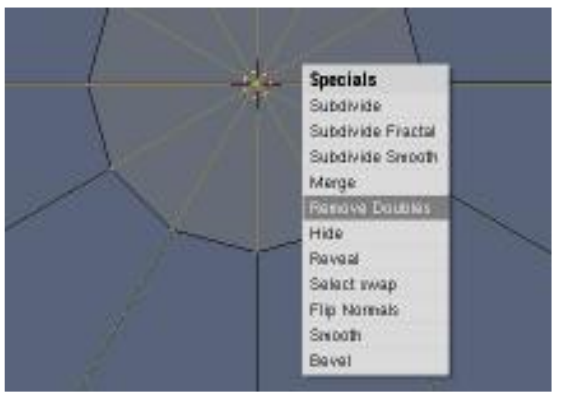


Fig27. Removing the doubles

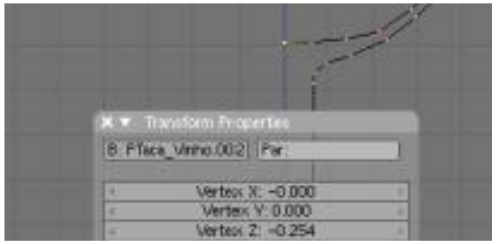


Fig28. Transform properrties

To remove doubles, press the [W Key] and in sequence [remove doubles] in the pop-up menu. (figure 29)

A new pop-up menu will show the number of vertices removed. In my example, there were 11 vertices removed. Now only one vertex remains at that location. Do the same with the circle at the top. This is just to demonstrate that a simple method could help us a lot. But, if we use it in an inadequate way, we may have more problems than we had before. All of this work, hard work, is absolutely unnecessary.

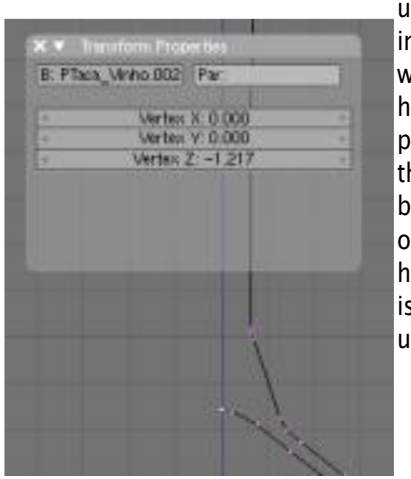


Fig29. transform properties

Now that we know that this problem could occur, we can modify our method with some planning. Let's see: going back to the very beginning, if we pay attention to the coordinates of the initial and end points when constructing our shape, and position them in a straight vertical line, we will eliminate the "tube-fication" on our model (that's the circular mess we just cleaned up). To get these two vertices aligned vertically, we have to use the Transform properties panel [N Key] as shown in figures 28 and 29. Change the values of the x, y or z coordinates as needed.

We could place the Center point to be aligned with the initial and end points however, I prefer to make the Center point coincident with the initial point. Select the initial point (vertex) of the shape, press [Shift+S] and in the pop-up menu, select the option [Cursor >> Selection] (fig 30).

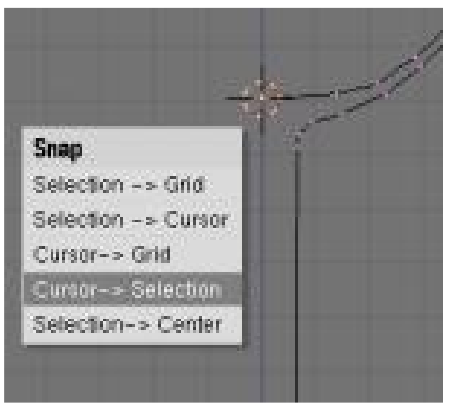


Fig30. Placing the cursor

Press the [Tab key] to exit the Edit mode. You will now be in the Object mode (you have to be in Object mode to use the Center Cursor feature). Now, go to the Mesh panel and click [Center Cursor]. (figure 31)

Fig31. JCenter cursor

Now, before you use the Spin tool, place the 3D Cursor at the center point of the shape.

Press [Tab] to leave edit mode, press [Alt+S] Figure 31 activating set smooth and then [Cursor >> Selection]

Fig32. Set Smooth

Press the [Tab key] to go back to Edit mode. Make sure all vertices are selected. Click the Spin tool button (make sure you're in the Top view).

Select all vertices again. Press the [W Key], and select Remove Doubles from the pop-up menu.

Go to the front view [Numpad 7]
Press [Z] to change to Solid mode.
Press [A] to select all the vertices of the shape
Press the [Set Smooth] button in the Link and Materials panel
(figure 32)

If you want a better look, you could use the Subsurf modifier found in the Edit buttons window. Add the Subsurf modifier and adjust the subdivision number to 2. See fig 33. Turn off the [Double Sided] button.

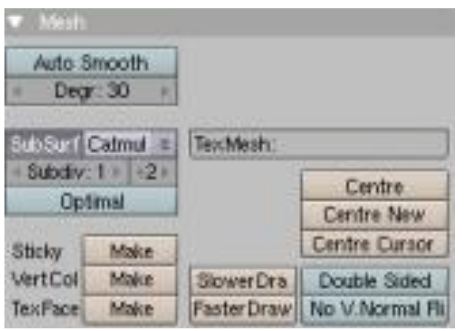


Fig33. The Mesh panel

If you see black lines in your model, you will need to Recalculate the Normals.

Press [Ctrl+N] and then select Recalc Normals Outside from the pop-up menu. The normals will be recalculated to point outside the mesh... see figure 34.

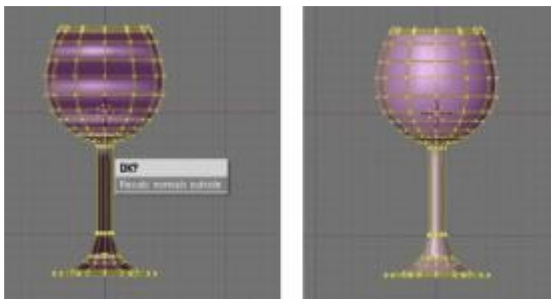


Fig34-35. Before and after normal calculation

Your wine glass should look like figure 35. Press [Tab] and take a look at your work. With this method, as you can imagine, can easily be applied to making a lot of kitchenware. We just construct the shape, place the start and end vertices in a straight vertical line and use the [Spin] tool to have a well-made mesh.

I hope this tutorial was helpful in showing examples of how things can go wrong. So, when you are constructing your model, make sure to plan things ahead. If something goes wrong, learn from the problem, and never give up.

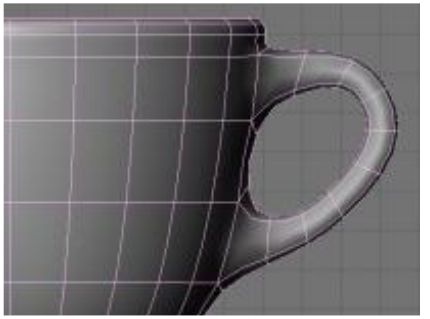


Fig35. Using extrusion we can create othe objects

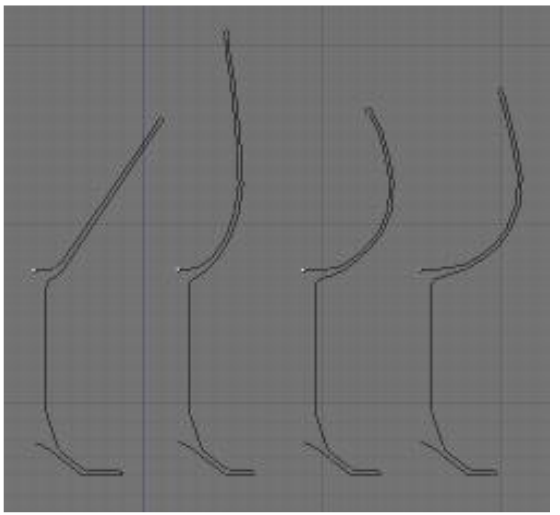
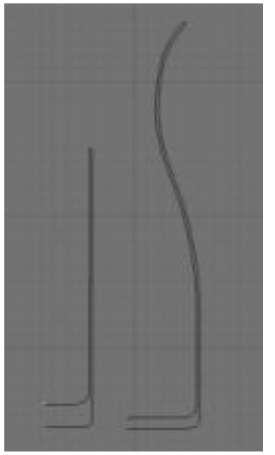


Fig36. Example shapes whih can me made by 'spinn tool'



UV MAPPING TECHNIQUES (LOW-POLY)

- By Roja

Level: Intermediate

Preparing a model for UV Mapping

UV mapping is a skill that takes a lot of time, practice, and experimentation to master. Once you get it, and understand it, it's really not very hard at all. It can be frustrating and take up way too much time if you don't do it properly or are lazy about it however.

What I'm going to explain here is how I go about preparing, or setting up a model for UV mapping. I will focus on a simple low polygon, symmetrical, game model. There are, however, many different ways to go about doing this, mainly dependent on what your model is.

The importance of planning

Before you even start to model, you should also be planning & thinking about UV mapping & texturing.

The reason for this is to save you time with UV mapping when you get to that stage. The best way to explain this is to give some examples.

Fig1. A lowpoly house with and without wireframe

Both UV mapping and texturing on this model was done even before the modeling was complete. Let me explain.

Take a look at the red colored wooden beams in the Fig1 , they are repeated all over the model and the windows are repeated too, as well as the sharp pointed polls sticking through the roof. If I had to modeled this entire house, normal method would have been to first model it then UV unwrap each repeatable part individually! Instead, I modeled 1 wooden beam, then I UV mapped that beam, textured it(although you can texture it at a later stage), then added it to the model, duplicating the objects as I needed.

Same thing for the windows and other repeating items in the model. This will clearly save you a TON of time.

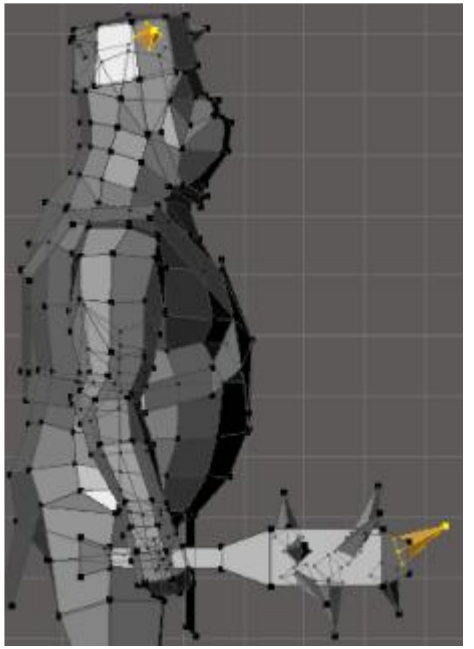


Fig2. A generic model

A second example is for smaller repetitive objects in your model. In this example the monster has a spiked club, and also a helmet with spikes on it. See the highlighted spikes in the image. Similarly there are a lot of spikes in the model, and it will take some time for you to duplicate all those objects, rotate them, and place them. Again the most optimal way to do this is to first model 1 spike.

Then UV map it, THEN duplicate and place it to finish your model. So again, you're UV mapping before! you have the model finished.

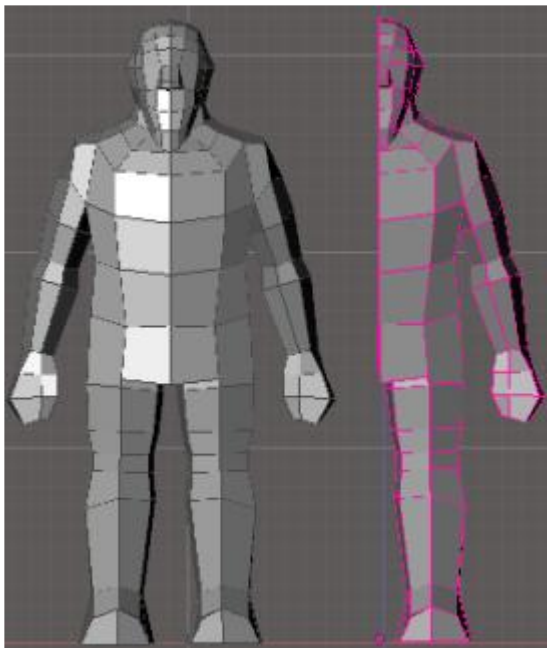


Fig3. A symmeteric model

Example 3 shows a totally symmetrical model, that I modeled using an instanced* copy. Since I only actually modeled half of it, and I plan on making the texture totally symmetrical, I should only UV map half of it! So before joining both halves of the mesh to finish the model, it's best to UV map it at this stage.

*Even though this is not a modeling tutorial, I thought it'd be important to note this: An instanced copy is a separate object that is changed in the exact same way as the original object is when you edit it. You can make an instance of your model by deleting half of it in edit model(basically splitting it in half symmetrically), then select the mesh in object mode, and press alt+d to instance it. This creates a duplicate that is an Instance. Then in Object mode, In the header bar click on Object->Mirror and choose an axis to mirror the mesh, and line it up next to the other. In current Blender releases you can use the Mirror modifier with similar results.

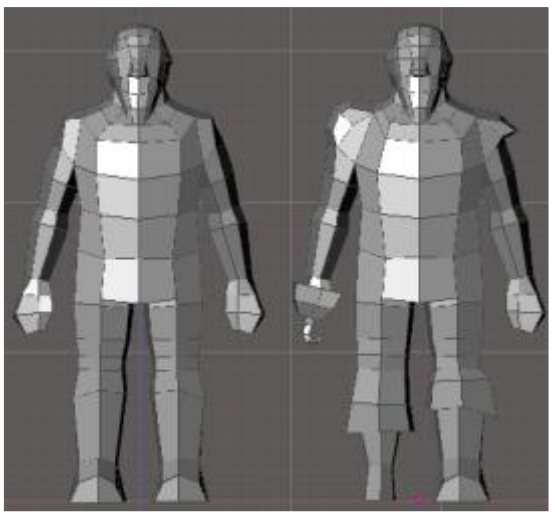


Fig4. Asymeteric model

A fourth example is dealing with partially(or fully) asymmetrical organic forms, such as a character model that has parts, or all, of the texture, or mesh asymmetric. The image shown below gives an example of a symmetrical mesh (left), and an asymmetric mesh (right).

As you can see, the asymmetric mesh has two different shoulder pads, 1 normal hand, 1 with a hook, and 1 normal leg with a boot, 1 with a wooden peg leg. For this type of model, it is best to make the complete model first.

Starting out

For the remainder of this tutorial I'm going to use a completely symmetrical model as an example. UV mapping on an asymmetrical model uses the same techniques, it just may take a little extra time. Here is a picture of my finished model, ready to be worked on for UV mapping:

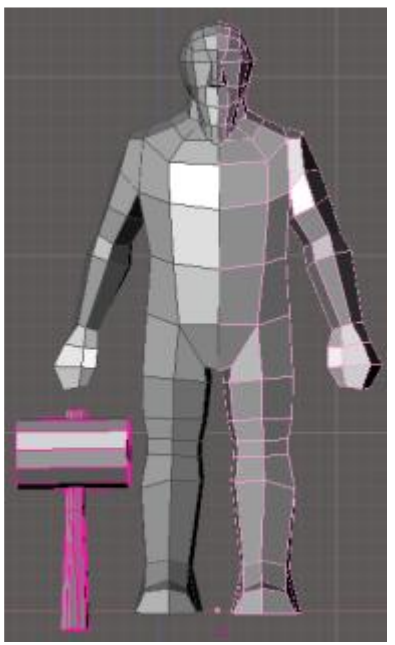


Fig5. Mesh ready for UV mapping

His body is completely symmetrical, as is his weapon. Right now there are 3 separate objects, 2 are the halves of his body, the other is the weapon. It's easier to keep the weapon as a separate object right now, and join it(select both meshes then ctrl+j to join) when you're done if you need it to be joined in the final mesh. We're going to focus on the body right now, so let's delete the one half and put the weapon on another layer, or hide it for now. So this is what we've got:

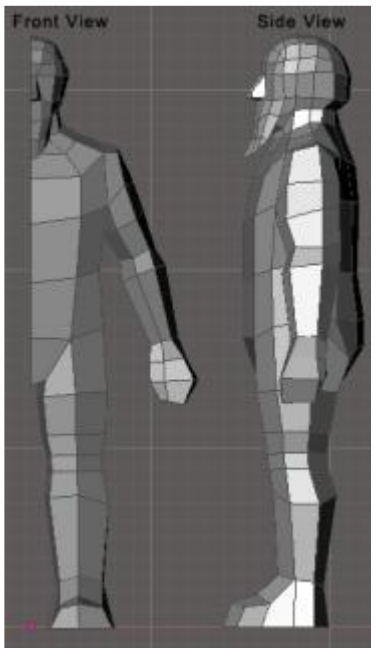


Fig6. The half part of the mesh

Seams

For an organic model such as this, I almost always use only 2 different UV unwrapping techniques found in the UV Calculations Panel: LSCM Unwrap, and From Window (Planar Mapping).

The LSCM Unwrap function is nice because it does a sort of automatic unwrap after you've defined seams on your model. I use the LSCM in conjunction with a planar mapping function because some parts of the model work best with one technique, other parts with another.

Seams are necessary for LSCM to work properly, but they also have one other really nice feature, and that is that they let you select pieces of your mesh in UV Face Select mode by pressing L and having your mouse cursor over or near the face. This will become invaluable when we get to the UV editing stage.

Ok, let's start defining the seams! We're going to start with a really easy part, the bottom of the foot. Start in object mode, press tab to go into edit mode, then make sure you're in Edge Select Mode, because we're going to select edges to make the seams.

Select all the edges around the bottom base of the foot. Hold [Shift key] as you select them so you can add edges to your selection. Then press [Ctrl + E] and a pop up box will appear.

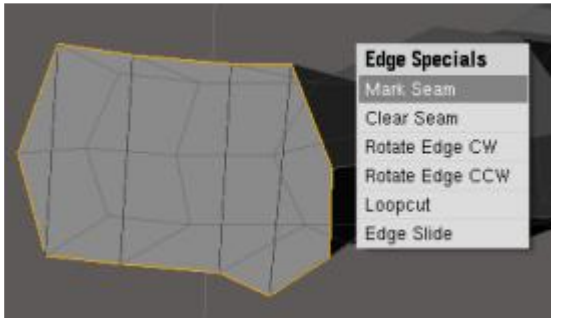


Fig7. Marking the seam

Click Mark Seam. Now you can de-select the edges, and a Seam edge line will appear. And that's all there is to making seams! You can also 'Clear Seams' incase you messed up or want to change them, do so by pressing [Ctrl + E] and choosing the Clear Seams option in the Popup. Moving along we're going to mark more seams throughout the entire model. This next picture shows a seam marked down inside of the leg, and around the waist.

Remember you should make your seams in places that are not directly visible in the scene, just in case you have a texture seam showing. You also want to put them in places that make it easy to break up the texture without causing visible seams(like a belt around the waist).

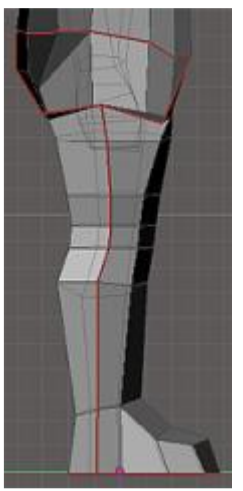


Fig8. Leg seams

Next is the arm I defined the seam around the shoulder joint, down and inside of the arm, and at the bottom of the hand as shown in the Fig9. Note that I also hid parts of the mesh(select areas you want hidden, then press [H Key]) to make it easier to click on the edges. To un-hide the mesh, press [Alt+ H].

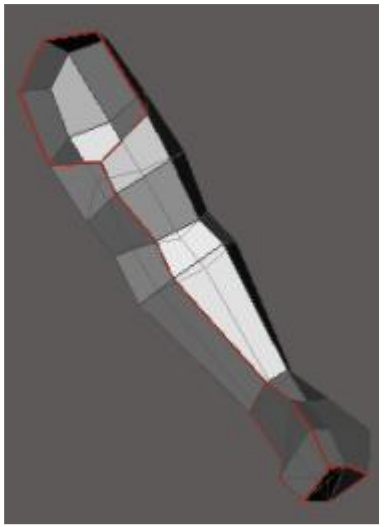


Fig9. Seams for arm

And lastly, the head. I went around the base of the neck, where his beard attaches to his face, and because I want to make him bald on the top of his head with some hair on the back, I put the seam on the back of his head to reflect that. I also put a seam around the face itself. Now we're done putting seams on his body. See fig 10.

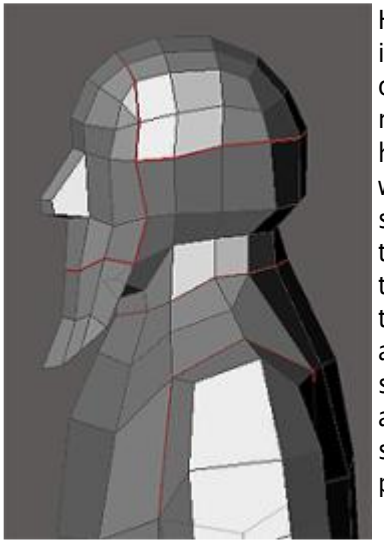


Fig10. Seams for head

Here's an image (Fig11) of the seams I made for his hammer weapon. I put seams around the circles at the ends of the cylinders, and also a seam going along one side of the poll.

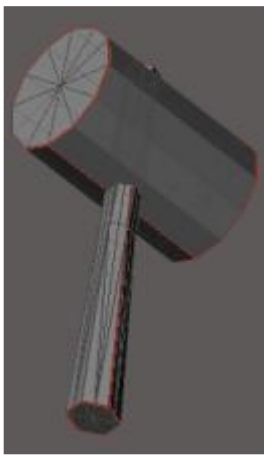


Fig11. Seams for hammer

And thats about it for the seams! Now lets move on to the actual UV projecting and unwrapping.

UV Unwrapping a 3D model using LSCM Unwrap and "From window" Planar projection.

Interface Setup

Before we begin, I'd like to mention that having a good workspace(or screen) set up in Blender with all the tools in your reach makes it a lot easier to work with. I made a custom screen setup in Blender just for UV mapping.

I have a 3d veiwport, the Action time line showing beneath that(in case I want to see the model in a different animation pose while texturing), the UV/Image editor window, and also the buttons window scrolled over to the UV Calculation menu(note that you have to be in the UV Face Select mode, and the Editing panel to see it). Here's what my UV mapping interface looks like:

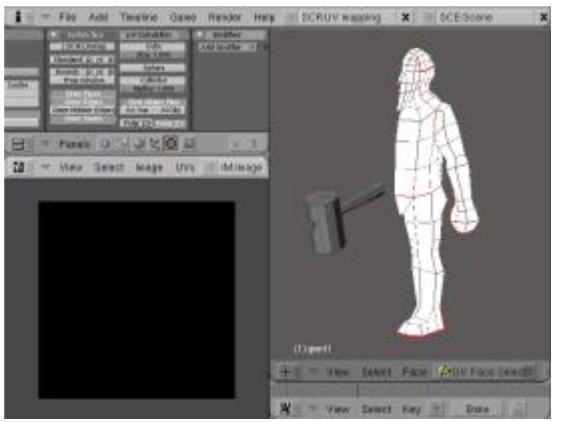


Fig12. The UV mapping interface

Using LSCM Projection Mapping

Select your model, and go into UV Face Select mode by pressing [F key], or by choosing that mode from the menu the 3d Veiwport panel. Your model will turn white. Make sure you have Draw Seams, Draw Edges, and Draw Faces all selected in the UV Calculations menu, like this:

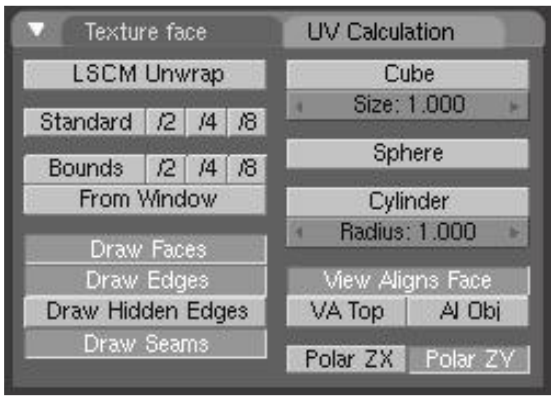


Fig13. UV calculation menu

Next, we need to make a new image in the UV/Image Editor. Click Image->New, then in the box that pops up type in a name if you want to, and also your texture size then press ok.

Fig14. UV image editor

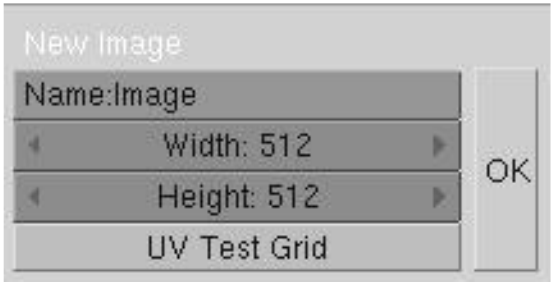


Fig15. UV image settings

Normal texture sizes for games are square and follow the powers of 2 i.e 64, 128, 256, 512, 1024. So choose the one that works best for you. As a side note, I find it best to work in a texture size that is bigger than what will eventually go in the game. So if my texture size is 256x256 in the game, I'll work with one that's 512x512, and resize it when the time comes. This is because it's easier to work with a larger size(as long as it's not too large), and it preserves more detail if you ever want your texture to be bigger.

A black box will pop up in the UV editor, you may have to zoom out a little to see the whole thing. Now, making sure you're still in UV Face Select mode with your model, press [A key] to select all the faces.

They should all become highlighted, and you'll also see some boxes pop up in the UV editor- those are the faces that we have to unwrap.

In the popped up UV Calculations menu, click on LSCM Unwrap. The LSCM Unwrap function uses the seams we created to unwrap the faces. You should get something similar to this, depending on your model of course:

Fig16. The LSCM umrapped UV

Now, as you can see, we've got some clean up work to do. We're going to have to do some work to resize, rotate, and move the pieces around to fill up the texture space and to lay them out as we want them.

The UV Test Grid

When UV mapping a model, you generally want to keep all pieces in proportion to each other as much as possible to prevent texture stretching. So a good way to see if parts are being stretched is to turn on the UV Test Grid. To do this, Click on (Image>> New), and making sure your texture size is correct, click on UV Test Grid and click ok. Now you'll see something like this:

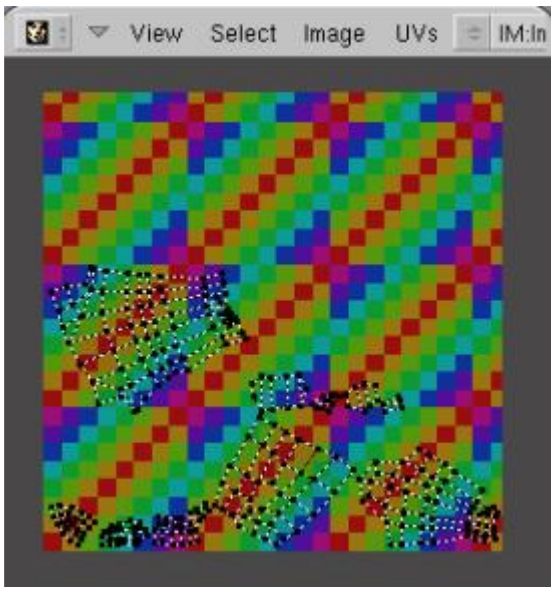


Fig17. UV test map

We also want to see this texture on our model. If you're not already in Texture mode in the 3d viewport, place your mouse over the 3d viewport and press [Alt+Z], or click on the Viewport Shading button in the header and click Textured. Now your model should look like this:

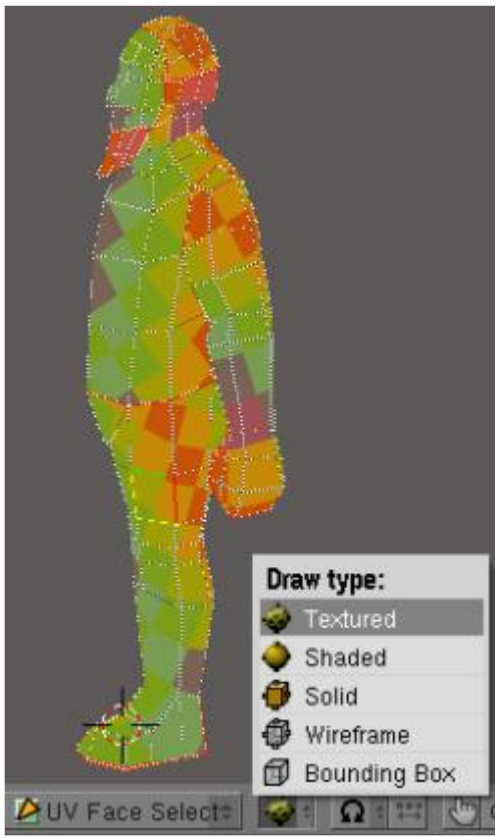


Fig18. Textured view of the mesh

Ok, now we can start to size the different UV pieces and see them updated on our model using the grid texture. The grid texture can be annoying to view the UV's properly at times, so you can always turn it off by making a new image texture and deselecting the UV Test Grid.

As you look at your model you'll notice that some of the grid blocks are larger than others, that's where the UV's are stretched, they don't accurately represent the proportions, so you'll have to make up for that by moving the UV vertices around as you work on unwrapping. Ideally, you want the colored grid boxes to be the same size across the entire model(or as close as possible).

Frequently Used Hotkeys

Working in the UV image editor uses a lot of the same hotkeys as modeling does. Press A to select all, A again to deselect. G moves the vertices, S scales, R rotates, B for the selection draw rectangle, M for mirror.

Also a VERY powerful hotkey is [L], for selecting an Element or Group of vertices(just hover your mouse over them and press [L]). You can also do this over your 3d model–and select an entire group made with seams as one example. Doing that makes it fast and easy to isolate areas you want to work on, without bothering the rest of the mesh or UV's

In addition to the hotkeys mentioned above to place the UV's into this position, I also use a few other functions to do different things. To place 2 vertices right on top of each other(like merging them), select them both, then press S then 0 (that is a zero). Or you can just Scale(S) them until they are as close as possible.

In the images below, you can see that I made the vertices of the front and back edges of the torso in a straight line. I did that by selecting all of them on the edge, then pressing S then X then 0. This aligns them all perfectly on the X axis. Substitue X for Y if you want them lined up horizontally.

Sometimes it's hard to tell what face lines up with the connecting face from another group. To do this, go into Active Face Select Mode by pressing C in the UV editor, or clicking Select->Active Face Select, then click on one face from one group, and you will see it becomes highlighted with 2 white edges, one green edge, and one red edge.

This highlighted face is also shown on the 3d model! So by selecting different faces you can see what lines up with what! This also makes it easy to select and move one, or a group of faces at a time, instead of picking up vertices.

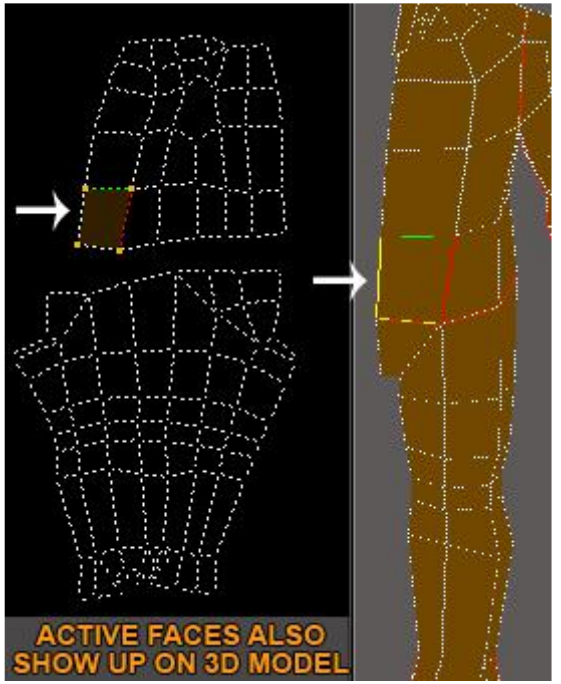


Fig19. Active faces

There are a lot of different functions that you should try out and play around with located in the UV/Image Editor drop down menus.

Unwrapping & Placing

I start off by working with the torso and leg UV's. I want to combine them together so that they will be seamless along the waist. As long as you're not wasting needed texture space, it's a good idea to combine 2 parts whenever possible.

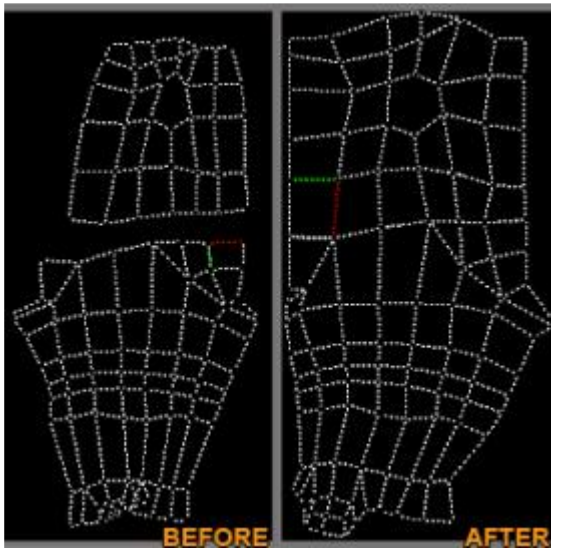


Fig19b. Unwrapping and placing

Moving on to the face, I'm not happy with how the LSCM unwrapper made it. I rather paint it straight on, as it's more natural that way. So what I'm going to do is to use Planar mapping for the different pieces of the head, in addition to the face, then attach as many edges to each other as possible.

Each piece needs to be isolated in the UV editor, and mapped one at a time. So first make sure all UV faces are deselected in the 3d viewport and UV editor by pressing A in the 3d viewport. Then, selectively press L on the group you want to start with, let's do the face. Only the vertices of the face will appear in the UV editor.

Hover your mouse over the 3d viewport and press 1 on the num pad to give you a front orthographic view(note that if you didn't position your model facing this direction it won't be a frontal view–you can change that by rotating your model so it faces the front). Now go to the UV calculations menu and click on From Window:

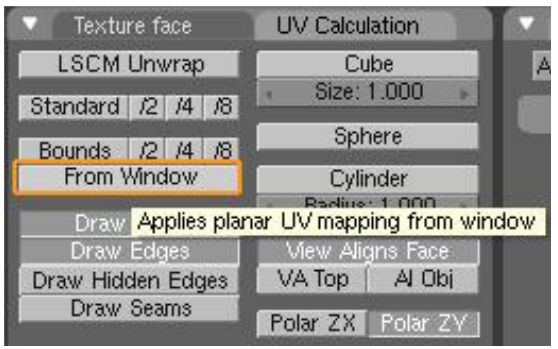


Fig20. selecting UV mapping

And below is another before/after picture showing the result of planar maping the face, top of the head, side of the head, and beard, then joining some edges together, and moving some vertices around(unwrapping) a little.

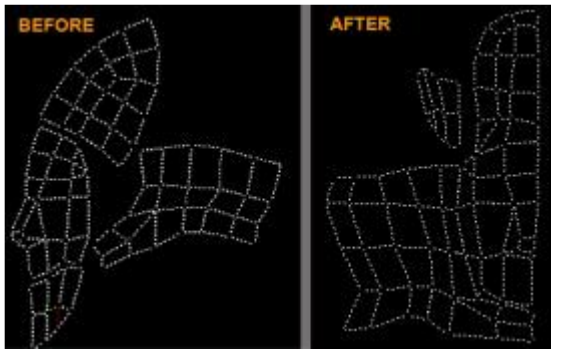


Fig21. Unwrapping for head and beard

For each piece I used a different view from the 3d viewport then I uv mapped it from that view–not all were done from the frontal view. So for instance with the top of the head, I first selected only those faces by pressing L in the 3d viewport, then on the num pad press 7 for a top down view, then I clicked on the From Window button. Press 3 for a side view(side of the head).

After doing a few more adjustments, and laying everything out with the least amount of blank unused space possible, this was my result: Fig22.

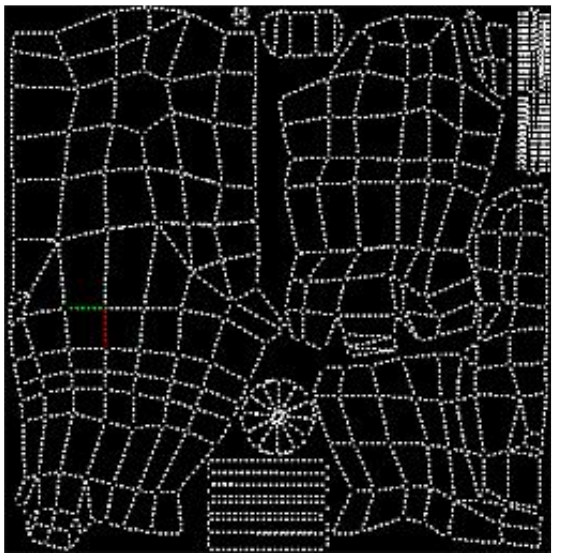


Fig22. The final Unwrapping

And now we're ready for texturing, the UV mapping is all done*! You can now duplicate the other half of your model and join the pieces together.

*As you texture the model, you may have to do some small tweaks with the UV's here and there, but try your best to keep them to a minimum unless there is something majorly wrong.

Using Blender's Texture Painting mode to create an outlined underbase.

Note: that this tutorial requires you to have a properly UV mapped model all ready to be textured.

Blender's Texture Painting features cannot replace a good 2D paint program, but it can be quite useful in blocking out important details that can be hard to find the right placement of such as eyes, muscle lines, etc. This way you'll be going back and forth between your 2D program and Blender a lot less, because you can get the placement right the first time!

So we start by having at least a 3d viewport with a model all ready to be textured, and the UV/Image editor open. Click on the button that looks like a pencil:

Fig23. Texture painting

Then Click on View->View Paint Tool.

Fig24. Paint tools

The Image Paint box will pop up: Here you can

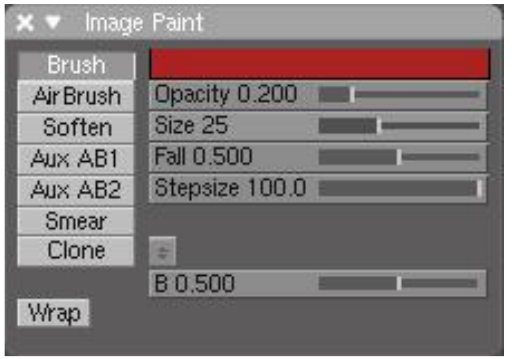


Fig25. Paint colour selection

choose the color of the paint, the brush, size, opacity, etc. Play around with the different features and paint strokes on the canvas to see what does what.

A few tips to note:
1. You cannot undo. If you press undo in another viewport, you will undo ALL painting you have done! If you press undo in the UV editor, nothing happens.
2. Right clicking in the UV editor in Paint model will select the color beneath the mouse-a very handy feature!
3. It's a good idea to turn off Draw Faces under the View menu of the UV/Image Editor. This is because the faces are slightly shaded, and if you right click on them to pick up the color that darker color will be picked up instead of the real paint color underneath.

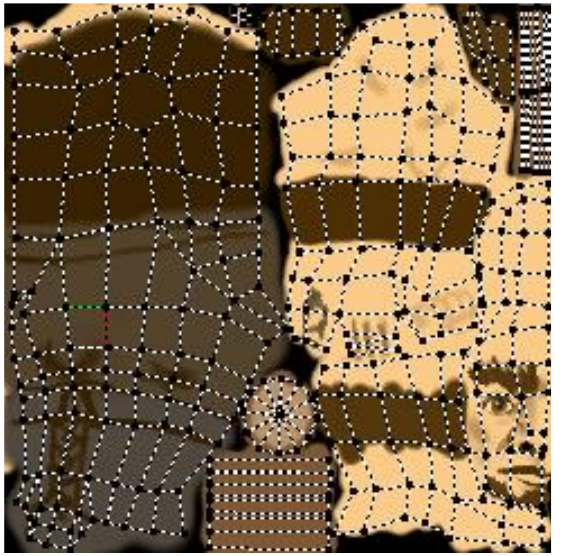


Fig26. UV map after painting

4. Make sure you have Update Automatically turned on in the view menu!

And here's what a really quick block in can look like:

Fig27. The UV texture after painting

You can certainly spend more time and detail on it in Blender. Since I plan on taking it to a 2D paint program, I didn't bother with making it look pretty, or getting the colors correct. Just enough to get the details I wanted placed. You can then click on Image->Save... to save your painted texture!

And here is what my model and texture look like completely finished:

*Note that I did use a 2D paint program to do the full texture, not Blender.

As a final note, I encourage all newcomers to experiment with the different tools Blender offers for unwrapping, as that's the best way to learn-to see what they do(in addition to reading the manual! ;) Don't be afraid to push buttons, just make sure you saved your file first!

-Roja

Final model

Final texture

USING BLUEPRINTS IN BLENDER

- By Edouard de Mahieu

Level: Intermediate

Using Blueprints in Blender

Modeling complex shapes and models in Blender, or in any other 3D application, may seem like a daunting task if you do not know where and how to start. How are you supposed to model a car with its complex hood or fender curves? This is where Blueprints come in. They will act as a modeling guide to help make the shapes you would normally have trouble making without blueprints. This article will start by explaining how to properly set up blueprint guidelines in Blender and then explain how to use them for modeling.

Set up

The first step is obviously obtaining blueprints to work with. If you do not have them already, then Google is your friend. You should also check out http://www.the-blueprints.com. We will be using the blueprints of a Ford Escort used in rally racing.

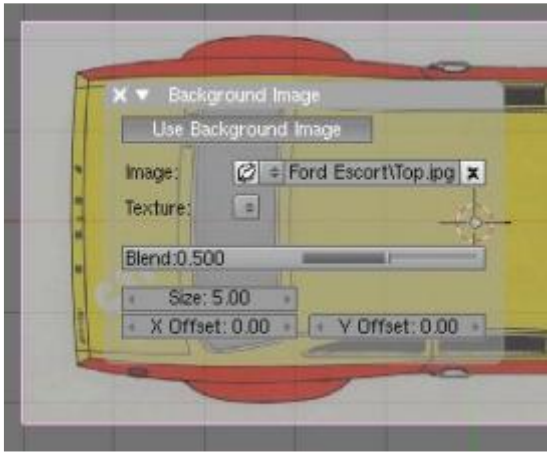


Fig1. Adding blueprint as background in Blender

The second step is dividing your blueprints into separate images. You must take special care to make sure that the sizes and dimensions of an image stay proportional to the other images otherwise you will find that they do not fit together in Blender.

This requires resizing the blueprints in the Gimp or Photoshop but will not be explained in this article. You should, after the modifications, have an image for each side/direction of your model, in this case, four (front, back, side, top). See fig2 blueprints with same dimensions.

Fig2. Blueprint from different sides

Once this is done, we need to set them up in Blender. The advantage of this setup is that you can have several blueprints on the same screen and be able to view them at different angles while giving you a better idea of depth and making it easier to model the pieces. To do this, start Blender and go to the top viewport and add the first image to the background by: *(View>> Background Image>> Use Background Image)* and select your image. Your first image, in this case, the top image, appears in the top viewport. Then add a plane: *(Add>> Mesh>> Plane)*. And scale to fit your background exactly using the **[S key]+[X key, Y key or Z key]**.

Now go to the material tab, add a new material and give it an appropriate name. Add a new Texture and load the same image as before. In the map input menu select UV. Split the viewport and open up the UV/Image Editor. Load up the same image. Select your object and go into the UV Face Select Mode.
Press the **[U key]** and select Cube from the menu. The plane UV coordinates will now appear in the UV/Image Editor. Position the vertices as before to map the image correctly on the plane. Make sure they fit correctly. Now go into the *(Draw Type>> Textured)*, using the **[ALT-Z]** shortcut. The texture will appear mapped on the plane.

Repeat with each of your images and position the planes correctly as to have a result similar to the following screenshot: Fig3.

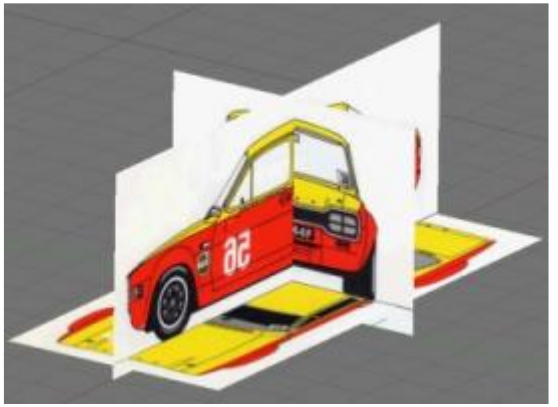


Fig3. Blueprints projected on plain mesh

You now have a strong blueprint setup that will ease and guide your modeling. Now, if this is done correctly, all that is left to do is to actually model. I will now try to explain, through an example, how to model the hood of the car using this blueprint setup.

Modeling the Pieces

Go into top view **[Numpad 7]** and *(Spacebar>> Add>> Plane)*. Place the first four vertices as seen in the screenshot. Fig4

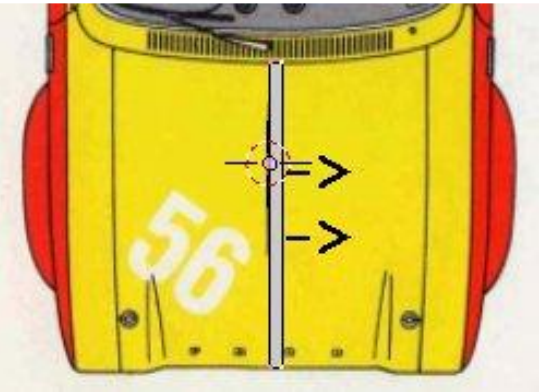


Fig4.Adding a plane

The first edge goes directly in the middle, as we will be using a Mirror modifier for the model. The second edge should be further to the right. We will be extruding this edge of two vertices, making small adjustments to their position to follow the blueprints. You should now have a mesh similar to the following screenshot:Fig5.

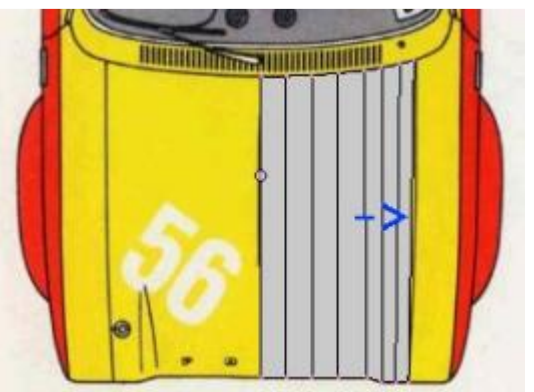


Fig5.Extruding the mesh

Now, go into front view and align the vertices with the front blueprint. Make sure that you align the correct vertice to the correct lines on the blueprints. You now have the general outline of the hood. Your mesh should now look like what you see in the screenshot: Fig6

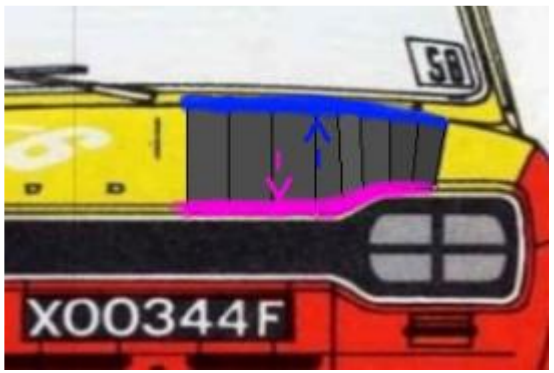


Fig6. Adjusting in the fron view

As pointed out by the blue arrow of the last top screenshot, the hood has a curve along its right edge. Subdivide the plane using Loop cuts: **[CTRL+R]** and use your mouse wheel to add several cuts at equal distances. Align the new vertices to the blueprints. The model of the hood is now essentially finished.

Conclusion

Continue using this technique along with the blueprints to complete your model. You should end up with a fairly correct reproduction of the blueprints in 3D. It will of course require small modifications and details that you should look up in reference pictures.

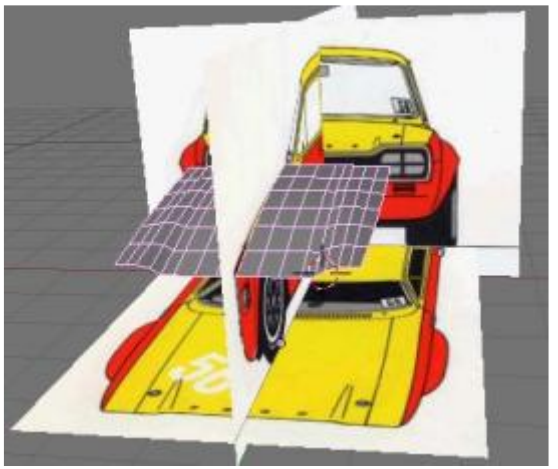
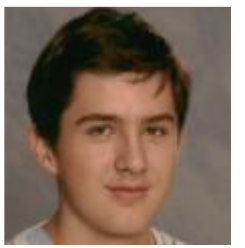


Fig7.Finished hood as seen in the setup

Most blueprints will give the general outline of your model but not every single tiny detail.

Happy Blending



Edouard de Mahieu A.K.A.
Edeehem or Slepnyrl

I live in Belgium and I am in my last year of high school. I have many hobbies such as modeling or tennis but computers are my main occupation.
I have been using Blender for about two and a half years now. I also spend a lot of time on the BlenderArtist forums trying to make myself useful and participating in the contests held there.

A SONOROUS TRIP

- By Diego Restrepo Paris

ABSTRACT

The main goal of this project was to create an immersive virtual environment. An environment where the user's body movements, as sensed by a dance pad, allow him or her to explore and develop the general atmosphere of its environment through the manipulation of a series of sonorous and visual cues. Secondary goal of this project was to test the flexibility of free software for creating a project of this kind. Blender was used to create the 3D environment and also as a real-time 3D engine. Csound and Audacity were used to experiment with sound generation and processing. This last goal was a very important personal one as I use free software because of my convictions: Thought must be open and flexible. Art is free and ideas must be available for anyone outside the artists' inner space.

INTRODUCTION

I always liked the games. My goal is to make artistic games based on the intellect. Some of the games with these characteristics are MYST, Monkey Island (1, 2, and 3), and Indiana Jones of Lucas Arts Films.

I think that games can help create new ideas for artistic projects. I began to enjoy the games combined with art.

But how can I create this project? My only basis is in my own experiences with the games and artistic projects in 3D.

My work is innovative because it is a game based on an environment made of artistic concepts and besides, because of that, is not a common artwork.

My pallet transforms in the textures and the different composition in the 3D space. The dance pad is innovative because it requires a new function; you use your feet to move in a virtual environment and find different sounds around you, every time that you begin the game, the experience will be different.

I use the traditional art forms of painting, sculpting and installations combined with synthetic sounds and digital graphics to make a new art object.

Why is this game artistic? Because I created a perception over the reality itself; the goal is to understand the dynamics of the game, that is why it is not easy to go from on world to another.

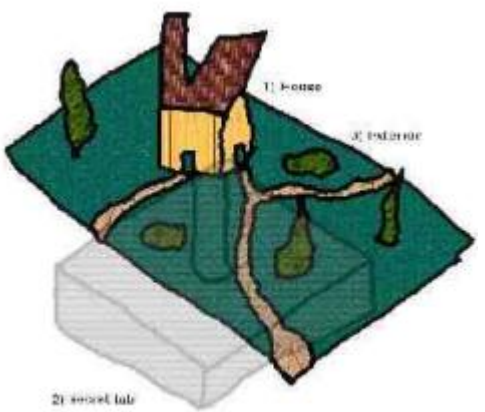
How did I get the idea of this project? I searched for a way to express different sensation in the body with reality, so I taught in virtual forms. I didn't want just a virtual 3D space to move around in, I wanted new ways to relate with the world through art. The idea was to make a 3D environment that involves the spectator.

A person approachs the world development by means of its relationship with sounds and objects. That's why the interface becomes so important; the interaction with the pad demands a corporeal force. This paper requires basic knowledge of the three mentioned software tools used to create the virtual environment.

CATEGORIES AND SUBJECT DESCRIPTORS

DESIGNING THE WORLD

All elements, artistic and synthetic, depend on each other. First, you need to have a good idea about the kind of artistic and synthetic concepts you want to express before the sound and texture libraries can be built.



All elements, artistic and synthetic, depend on each other. First, you need to have a good idea about the kind of artistic and synthetic concepts you want to express before the sound and texture libraries can be built.

For this project, the concept in all maps is to travel across the experimental virtual 3D environments with synthetic sounds.

[1] Exterior = express liberty, freedom; strange language from a different outside world.
[2] House = articulates ideas about family, community, protection, progress and intelligence.
[3] Secret Lab = communicates metallic sounds, aggressive, the dark-side world.

TEXTURE

I use the concepts of the Exterior, Secret Lab and the House to create the textures in the pictures using GIMP 2. When you save the final image, it is important to convert it to JPG format so it uses less memory in the game engine.

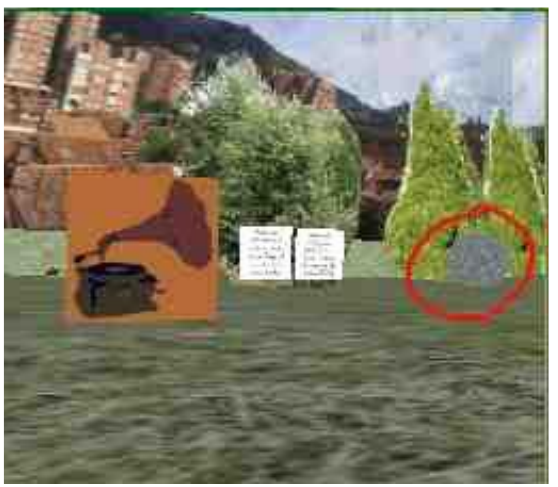
The textures of each level had to express the concept of the environment. If you need to make a realistic environment you need a realistic texture. If you want a fantasy look, you need textures with fantasy concepts. Otherwise, it is a good combination to mix reality with the fantasy; it results in an eye-catching contrast.

EXTERIOR

First world

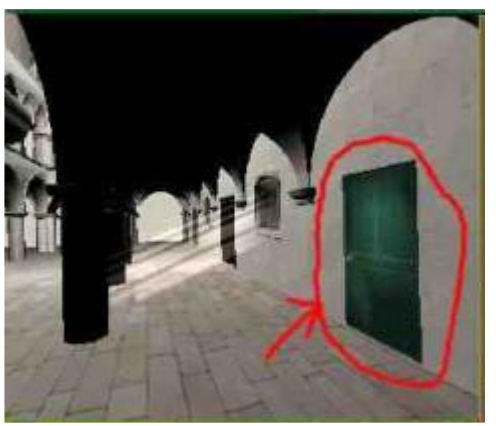
There, I imagined a golf course that has a natural, fantastic game ambience. I based it in the theory of Pythagoras of the stars sounds; he said that one sphere produces different sounds in the universe. The ball is used in this level to produce sounds.

Also, Plato said that the stars produce a harmonic sound and this sounds are true in the game environment. The environment has a big ball and you need to push the ball around in the world, the ball smashes on the objects producing different sounds.



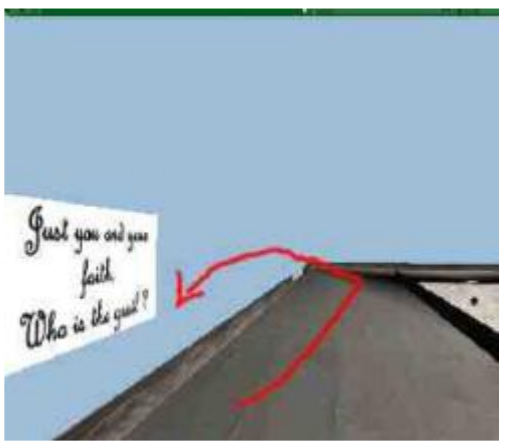
You pass to the next world after you have smashed the ball into a golf hole.

This world is based on the atmosphere of a Colombian colonial house. This house has three levels: the first level has doors, when you touch them, constant sounds are produced so you can compose music using different orders of touching the doors.



On the second floor, the pictures make the environment; you need to find the pictures using the clues of the different paragraphs that are all around the place.

The third floor is a suicide prove based in faith, you need to decide to go down to the house. I based it on the movie, Indiana Jones (1989).



SECRET GRAIL: Third world

In this level, you have to find the real grail cup and listen to the environment. The grail is made of wood because Jesus is a carpenter and normally the cup is of wood. I liked the concept taken from the movie because it expresses authentic approaches to the world's development by its relation with sounds and objects. That's why the interface becomes so important; the interaction with rug demands a corporal force.

CONCLUSIONS & FUTURE WORK

You can create an artistic project with a 3D environment, using free software. I worked with concepts about sounds and imaginary environments. It is important to make a plan of what you want to do. But what is important when you make the project, is to generate the ideas during the building process.

In this way, you can enrich your process creatively with your own stuff.

ACKNOWLEDGMENTS

I would like to thank the following persons for their support in making this project a reality:
C. PinzÃ³n, Technical and game engine support.
L.M. Barrero, (writing help).
D. Prieto and D. Doria sounds and music.

REFERENCES

http://www.etc.cmu.edu/curriculum/bvw/
http://www.blender.org/cms/Home.2.0.html
http://www.its2.uidaho.edu/cti/
http://www.csounds.com/
Gioseffo Zarlino, Harmonices Mundi.
AristÃ³teles, Del cielo .
PitÃ¡goras, La RepÃºblica

ID 172609Tras 16#133-30 int 34
BogotÃ¡Colombia.

+(057)4832375, Colombia
arcanouv2@hotmail.com
dieg-res@uniandes.edu.com

Diego Restrepo Paris

Diego Restrepo París is a 24 year old art student from "Universidad de losandes" in Bogotá. You can contact him at dieg-res@unaindes.edu.com

Craig Robinsion - Sphynx Crate

Bullix - To the competetion

Bullix - Uni Koxx

Jesse Nelson - Bismark

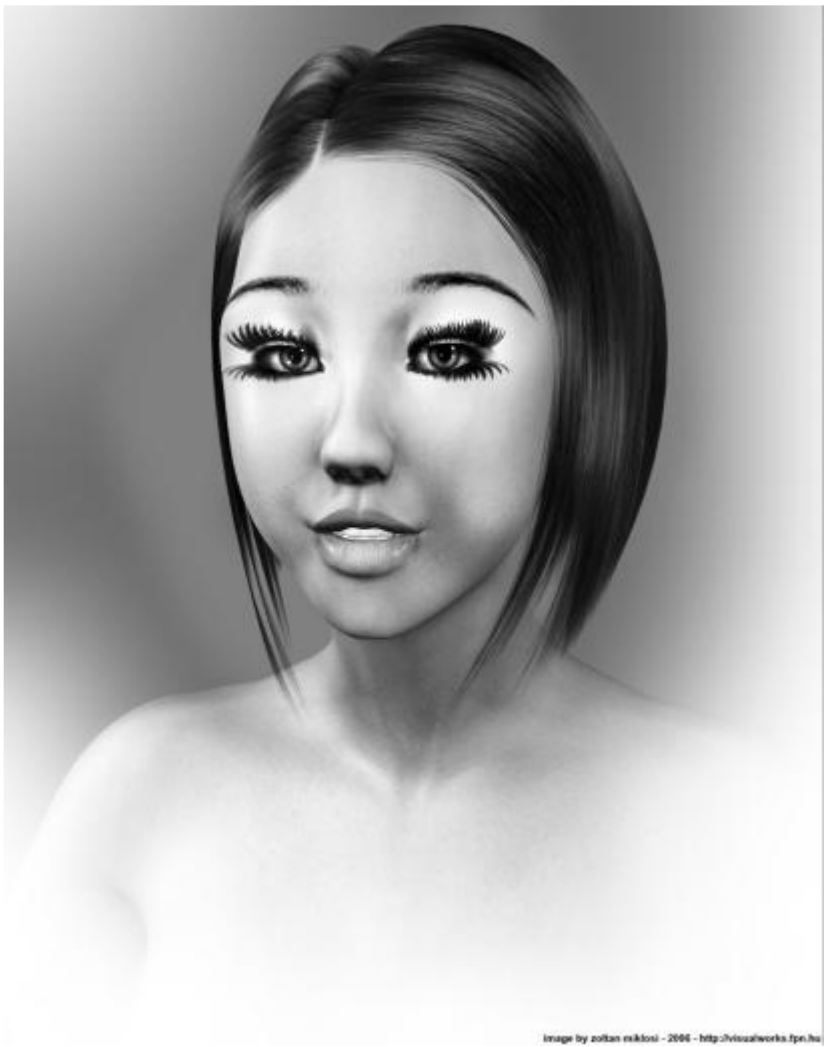
Jesse Nelson - Obi F1

Rogerio Pedriz - Mill's Zone - Orion Tear

Edouard - Sidi Bou Said

Edouard - Aira comet

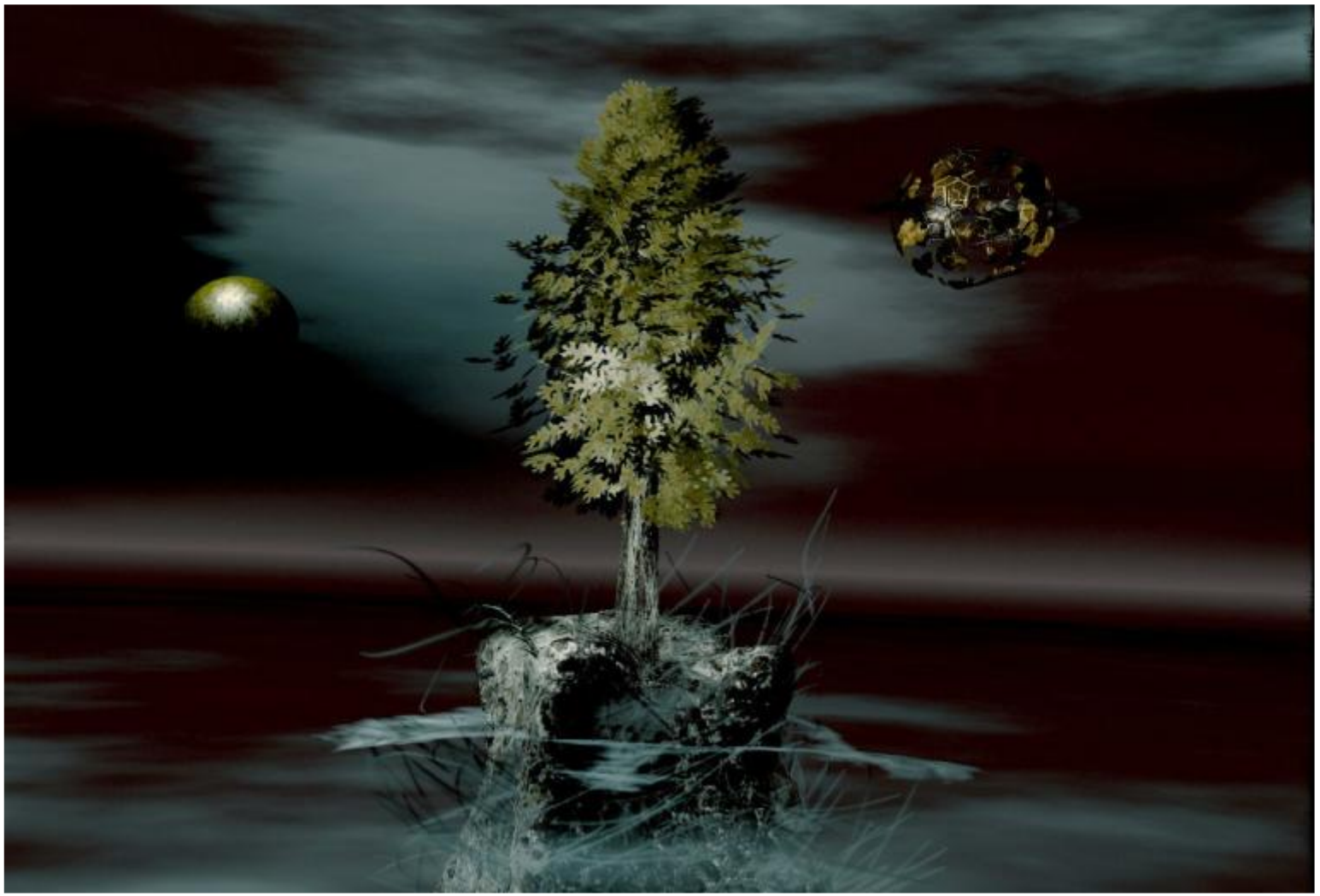
Zooly - The Train



Zooly - Potrait and Asian girl

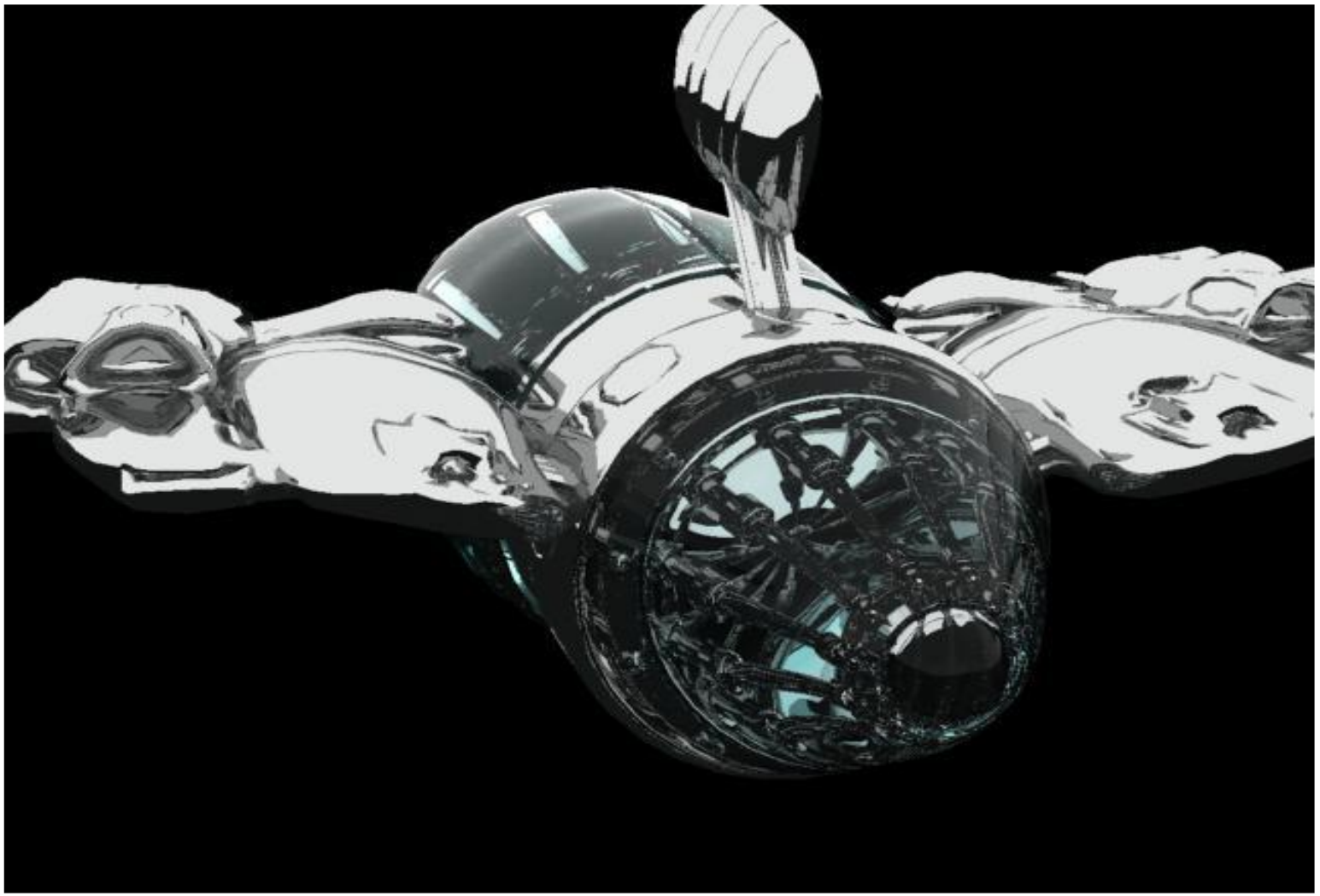
Zooly - Cool

Salavador Gracia Bernal - Chilmespers

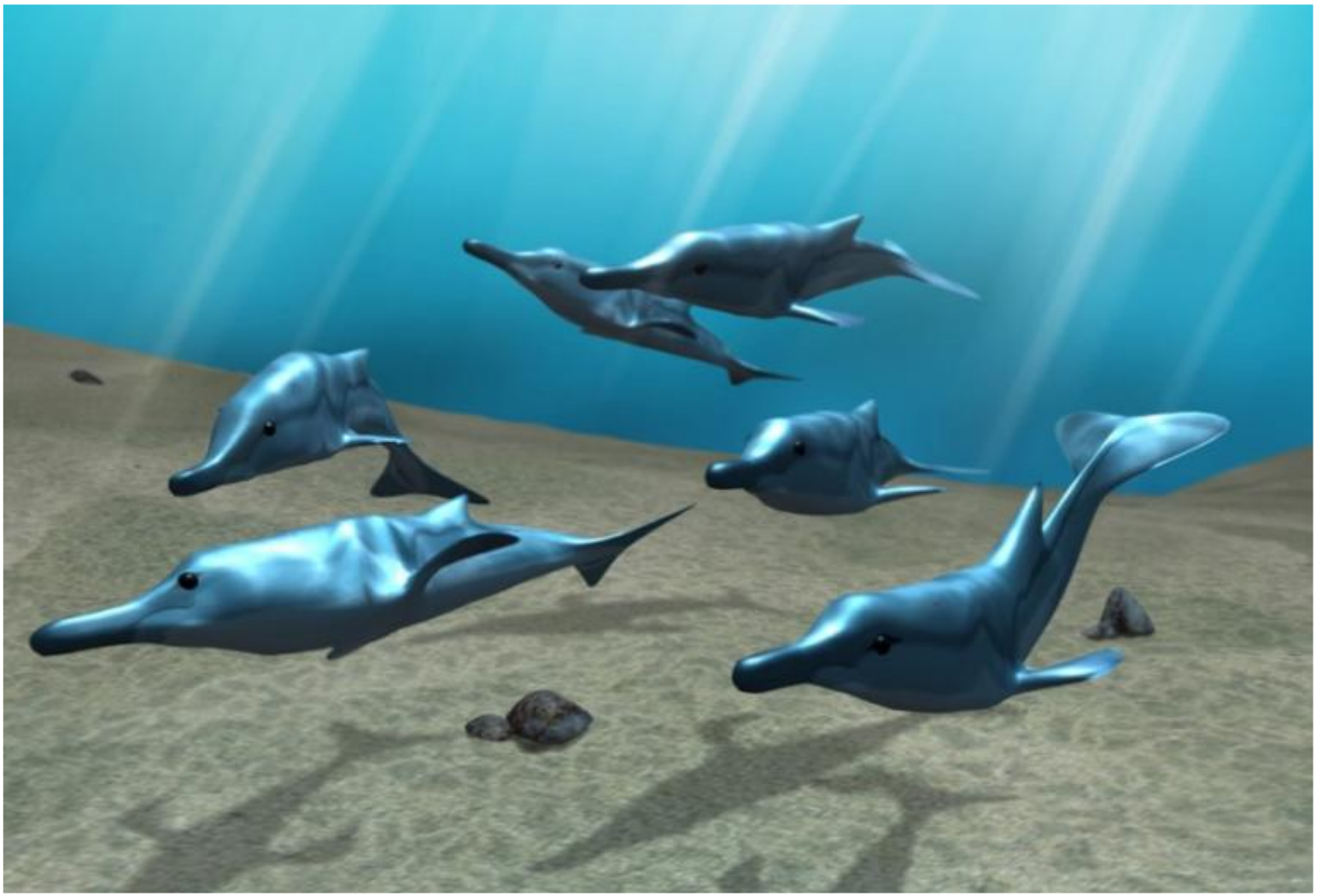


Mariano - A lost tree

Dia'a el khawaldeh - Red horizons



Dia'a el khawaldeh - Satelite

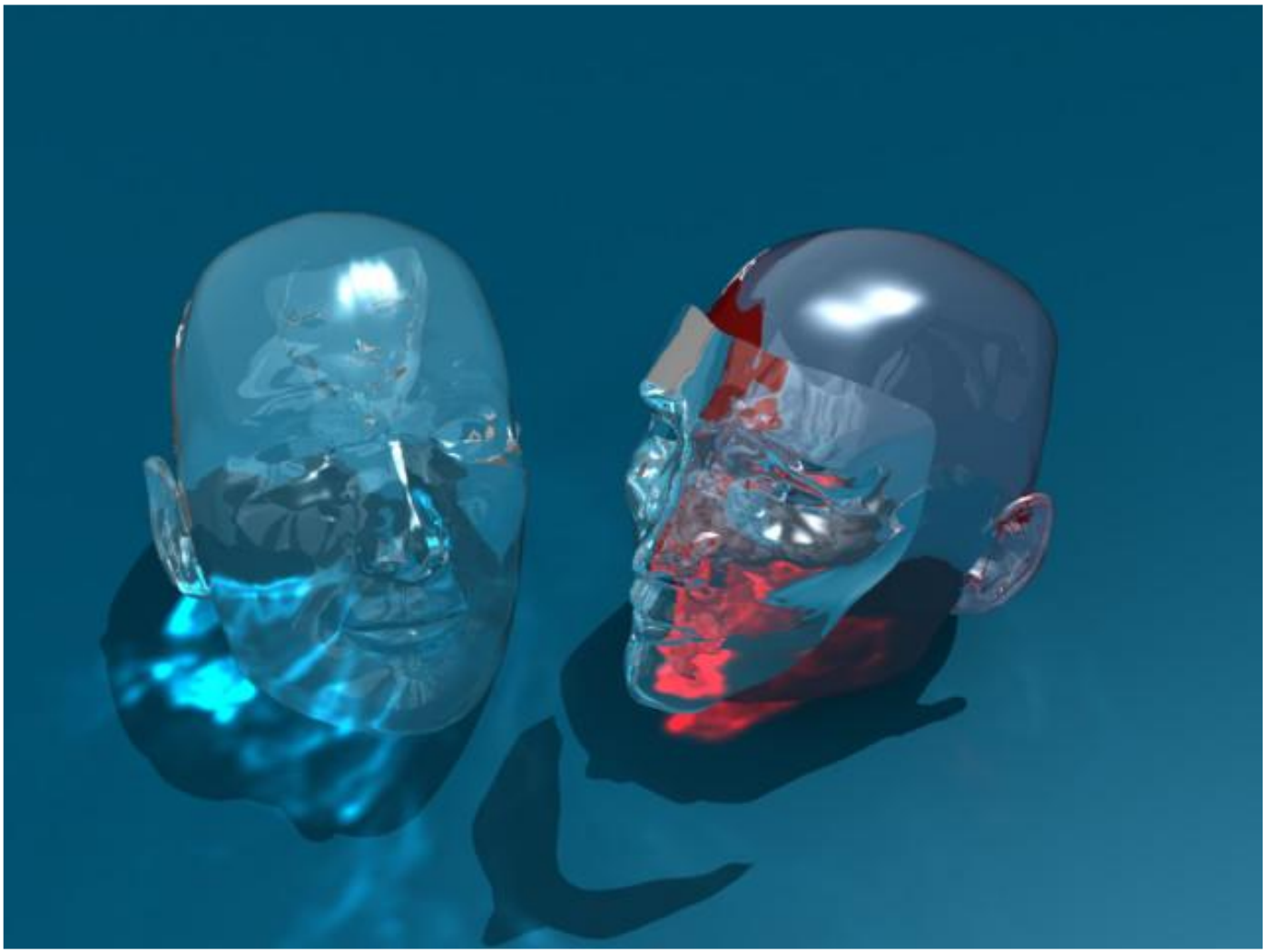


Alhaitham Jassar - Dolphins

Dhdewey - Ghoul

Jayson Allen Bayne - Robot

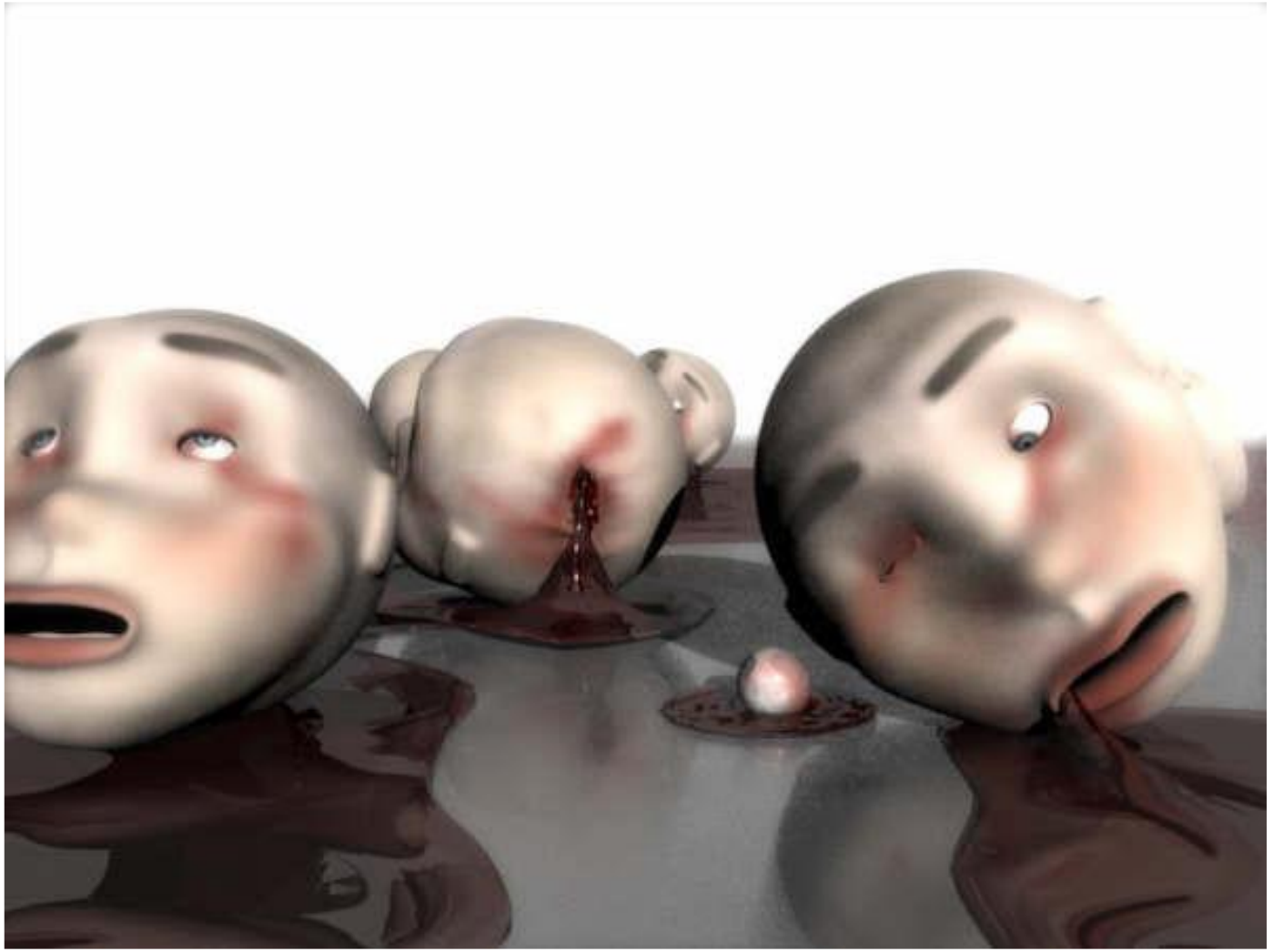
John Starr - Glass of water



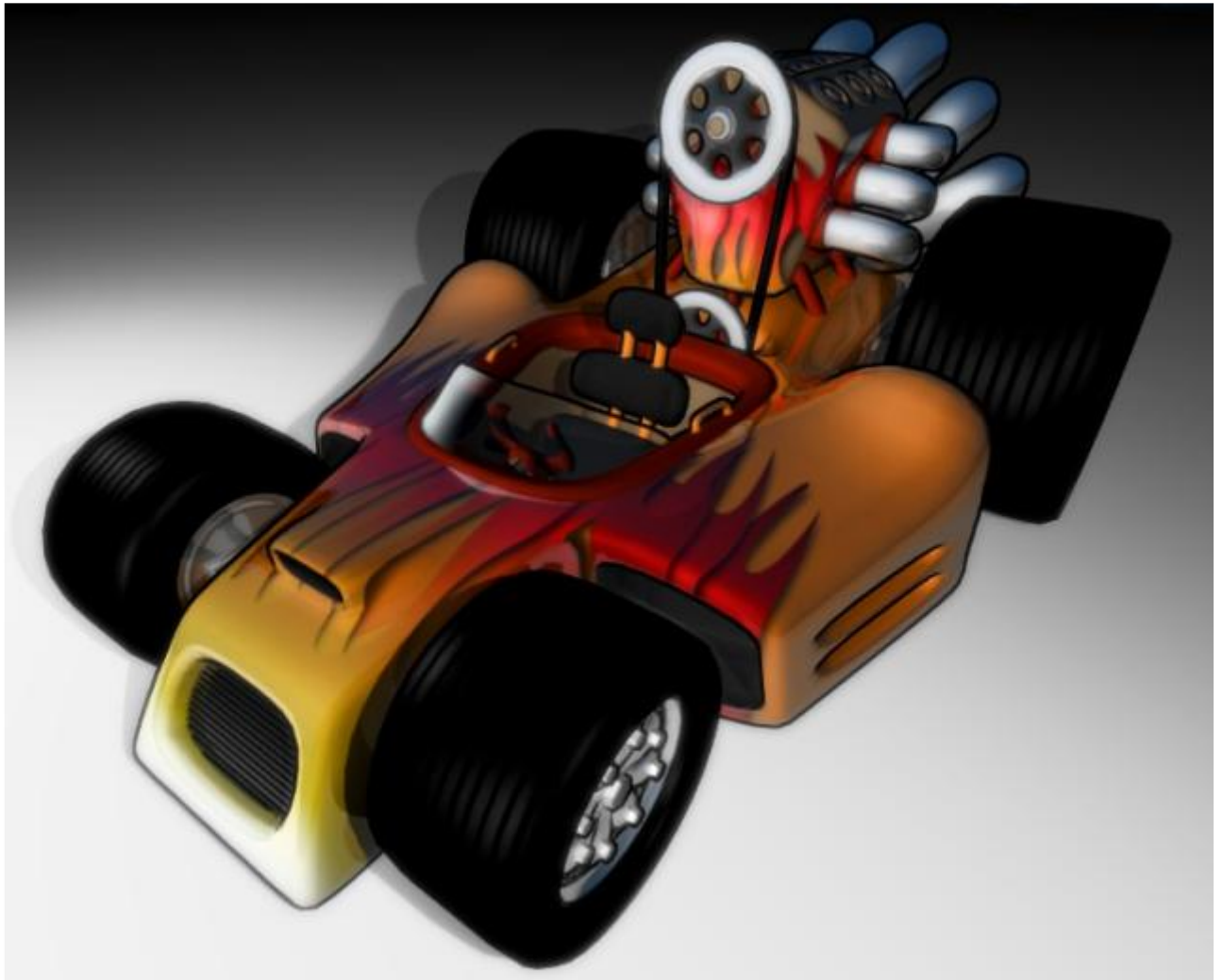
Fabien Sanner - Tetec

Hayes Nathaniel Dean - Dead snow

James Partaker - Ferrari 360 Modena



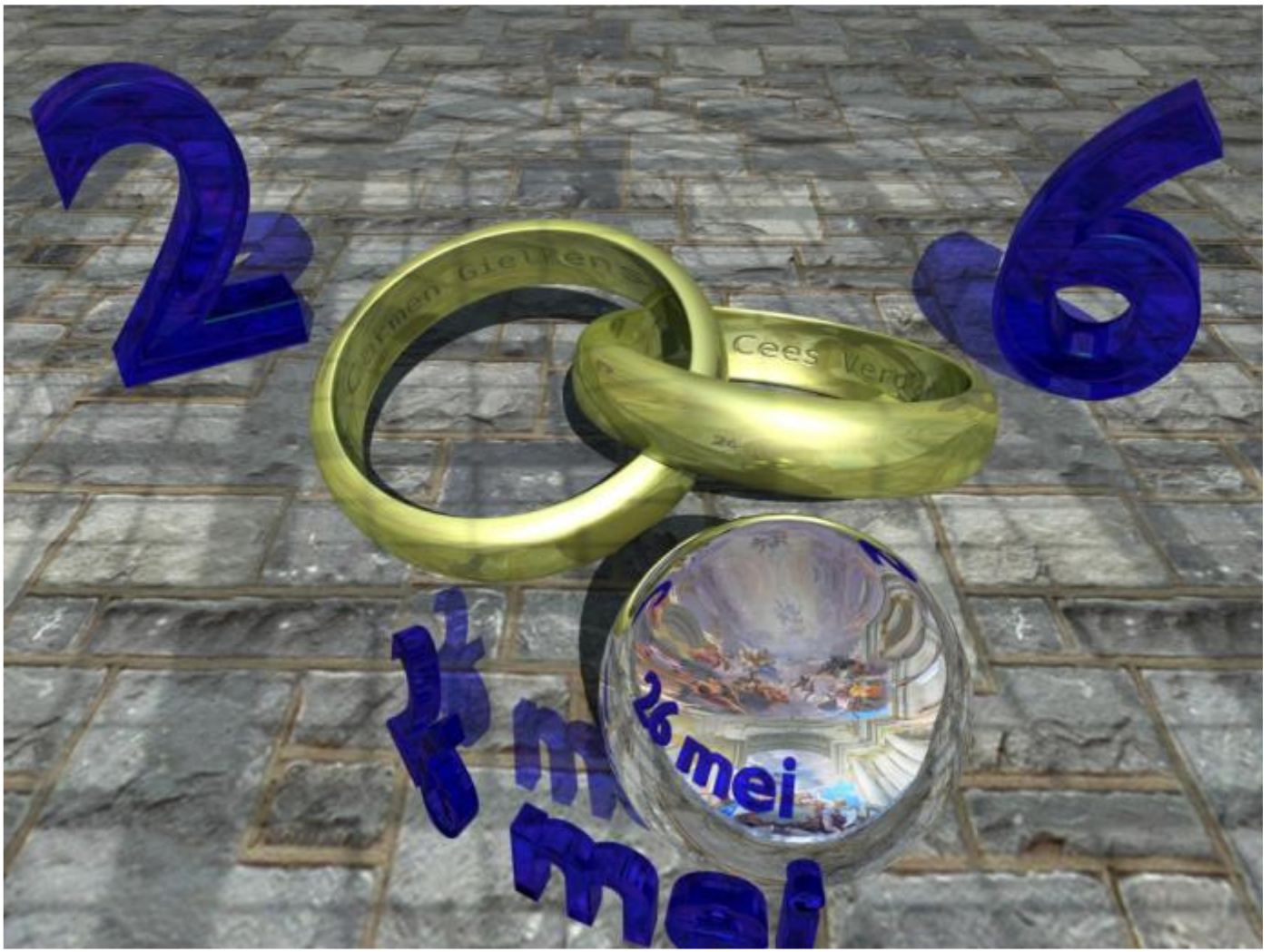
Lesse burntse - Headshot



Nivik Masterton - GoCart

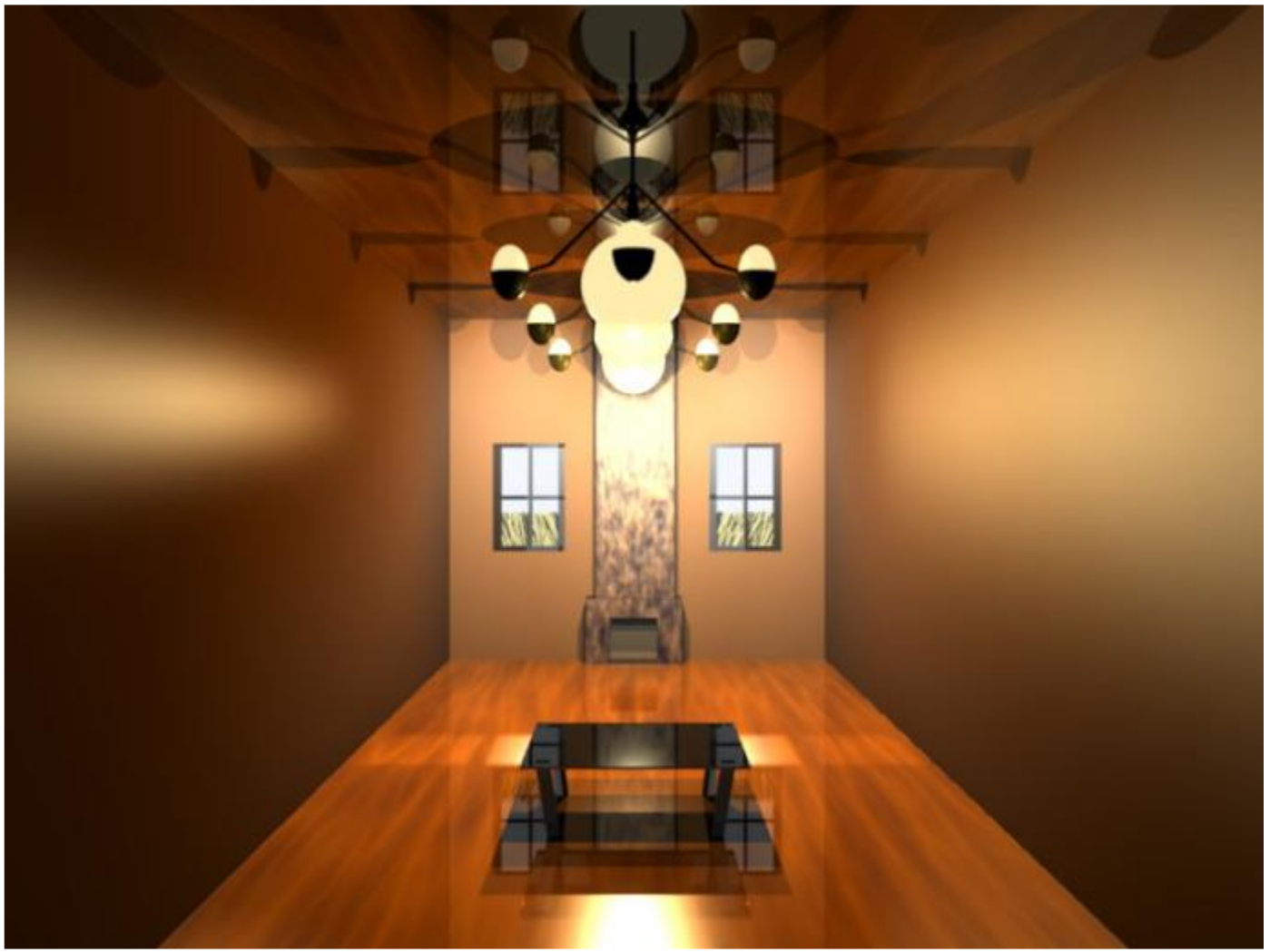
Pavel Shabanov - The cars

Ramon cerrols - Urchopia

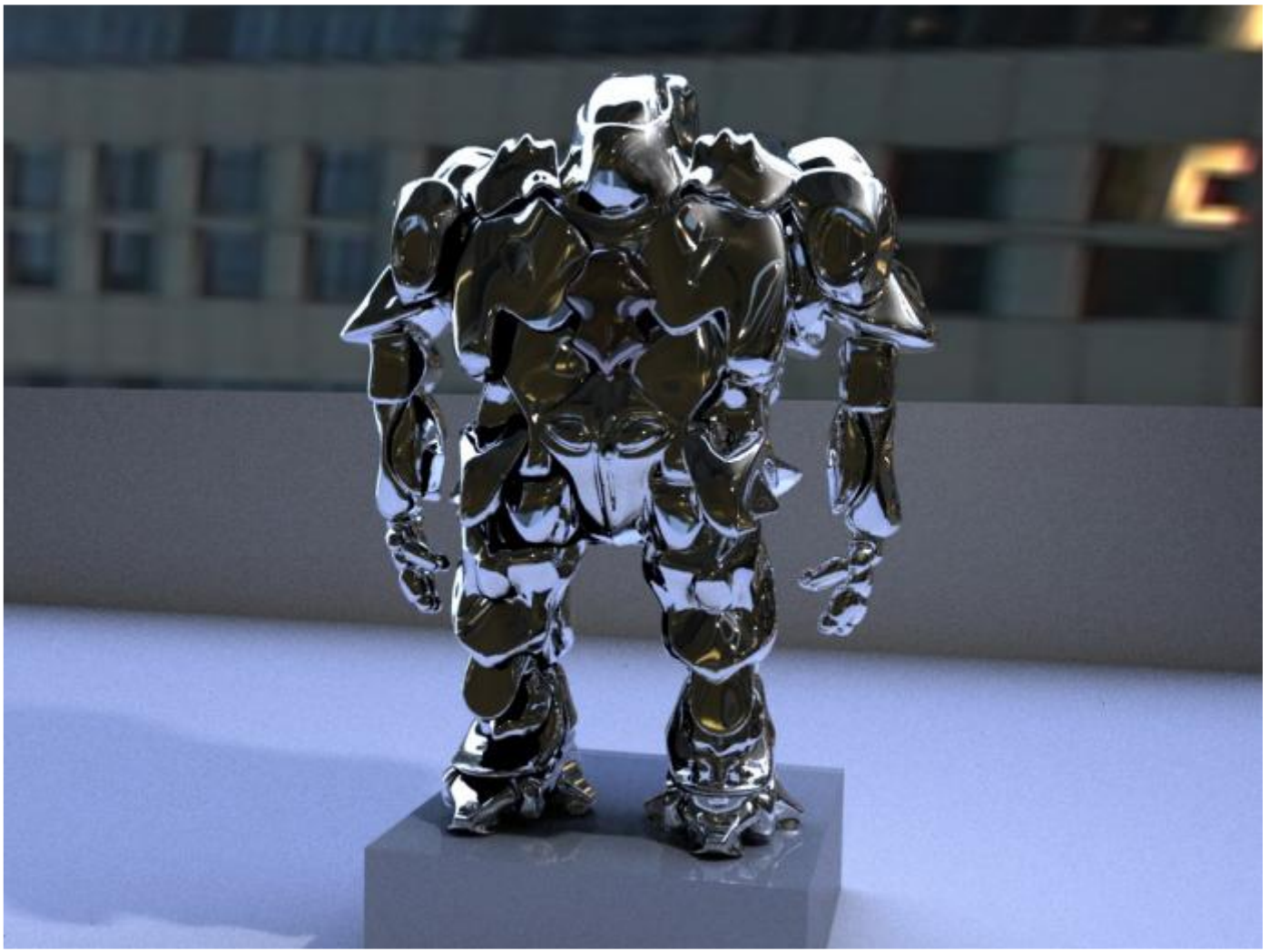


Serge Gielkens - Joint

The Cytron - Meet Spike

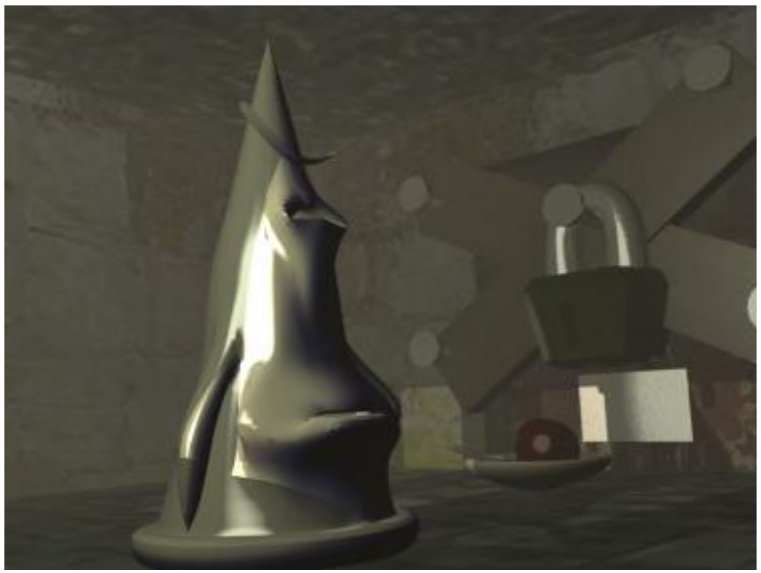


Thisara Dhananjaya - The wooden hall

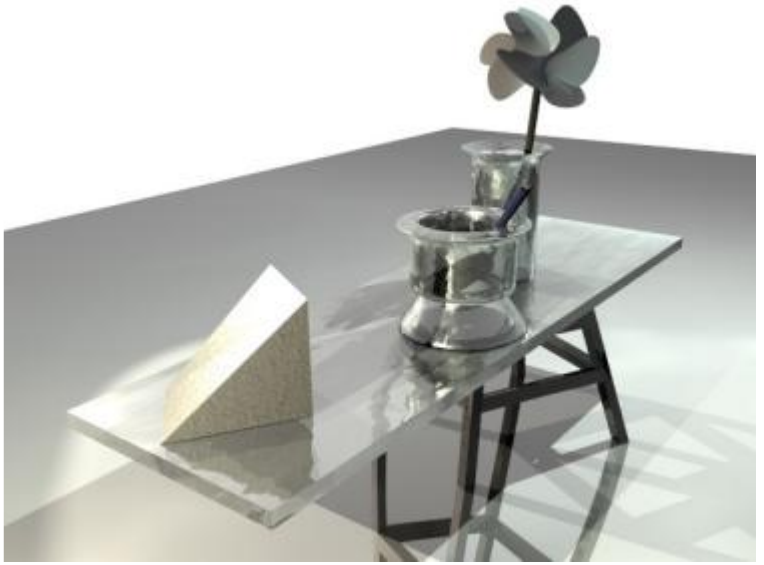


V Gostyax - U Skazki

Lizara - Carapelosojos



Lizara - Carcel



Lizara - Ejersico & Plane

Upcoming Issue

Theme : Blender in Architecture and Games

Blender's Game Engine explained
Blender in Architecture articles
Blender 2.42 Features

Disclaimer